

Lecture Notes in Social Networks

Tansel Özyer *Editor*

Social Media Analysis for Event Detection

 Springer

Lecture Notes in Social Networks

Series Editors

Reda Alhaji, University of Calgary, Calgary, AB, Canada

Uwe Glässer, Simon Fraser University, Burnaby, BC, Canada

Advisory Editors

Charu C. Aggarwal, Yorktown Heights, NY, USA

Patricia L. Brantingham, Simon Fraser University, Burnaby, BC, Canada

Thilo Gross, University of Bristol, Bristol, UK

Jiawei Han, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Raúl Manásevich, University of Chile, Santiago, Chile

Anthony J. Masys, University of Leicester, Ottawa, ON, Canada

Lecture Notes in Social Networks (LNSN) comprises volumes covering the theory, foundations and applications of the new emerging multidisciplinary field of social networks analysis and mining. LNSN publishes peer-reviewed works (including monographs, edited works) in the analytical, technical as well as the organizational side of social computing, social networks, network sciences, graph theory, sociology, semantic web, web applications and analytics, information networks, theoretical physics, modeling, security, crisis and risk management, and other related disciplines. The volumes are guest-edited by experts in a specific domain. This series is indexed by DBLP. Springer and the Series Editors welcome book ideas from authors. Potential authors who wish to submit a book proposal should contact Annelies Kersbergen, Publishing Editor, Springer e-mail: annelies.kersbergen@springer.com

Tansel Özyer
Editor

Social Media Analysis for Event Detection

 Springer

Editor

Tansel Özyer
Department of Computer Engineering
Ankara Medipol University
Ankara, Turkey

ISSN 2190-5428 ISSN 2190-5436 (electronic)
Lecture Notes in Social Networks
ISBN 978-3-031-08241-2 ISBN 978-3-031-08242-9 (eBook)
<https://doi.org/10.1007/978-3-031-08242-9>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

| | |
|--|-----|
| A Network-Based Approach to Understanding International Cooperation in Environmental Protection | 1 |
| Andreea Nita and Laurentiu Rozylowicz | |
| Critical Mass and Data Integrity Diagnostics of a Twitter Social Contagion Monitor | 19 |
| Amruta Deshpande, Vladimir Barash, Clayton Fink, Christopher Cameron, Aurora Schmidt, Wei Dong, Michael Macy, and John Kelly | |
| TenFor: Tool to Mine Interesting Events from Security Forums Leveraging Tensor Decomposition | 57 |
| Risul Islam, Md Omar Faruk Rokon, Evangelos E. Papalexakis, and Michalis Faloutsos | |
| Profile Fusion in Social Networks: A Data-Driven Approach | 89 |
| Youcef Benkhedda, Faical Azouaou, and Sofiane Abbar | |
| RISECURE: Metro Transit Disruptions Detection Using Social Media Mining And Graph Convolution | 111 |
| Omer Zulfiqar, Yi-Chun Chang, Po-Han Chen, Kaiqun Fu, Chang-Tien Lu, David Solnick, and Yanlin Li | |
| Local Taxonomy Construction: An Information Retrieval Approach Using Representation Learning | 133 |
| Mayank Kejriwal, Ravi Kiran Selvam, Chien-Chun Ni, and Nicolas Torzec | |
| The Evolution of Online Sentiments Across Italy During First and Second Wave of the COVID-19 Pandemic | 163 |
| Francesco Scotti, Davide Magnanimiti, Valeria Maria Urbano, and Francesco Pierri | |

| | |
|--|-----|
| Inferring Degree of Localization and Popularity of Twitter Topics and Persons Using Temporal Features | 183 |
| Aleksy Panasyuk, Kishan G. Mehrotra, Edmund Szu-Li Yu, and Chilukuri K. Mohan | |
| Covid-19 and Vaccine Tweet Analysis | 213 |
| Eren Alp, Bedirhan Gergin, Yiğit Ahmet Eraslan, Mert Can Çakmak, and Reda Alhajj | |

A Network-Based Approach to Understanding International Cooperation in Environmental Protection



Andreea Nita  and Laurentiu Rozyłowicz 

Abstract Environmental international treaties and multilateral agreements are complex arenas for cooperation between parties with the overarching goal of promoting our society’s environmental sustainability. International environmental protocols or treaties can be seen as communities that work together to achieve common environmental goals. To analyze the dynamic landscape of international cooperation in environmental protection in the last decades, we compared the cooperation established between the international parties that ratified the most important environmental treaties by continent and assessed the implications for transboundary environmental issues. Network analyses uncover clusters of the most popular and important international environmental treaties and the most influential brokers at the continental level in promoting environmental sustainability across borders. We further analyzed the structural patterns in international environmental protection to identify key promoters of cooperation, most collaborative countries, and most important environmental issues tackled by international agreements by considering the party’s continent as an attribute. Given the growing environmental issues over the past few decades—including the effects of climate change, future strategies must consider missing elements leading to improper environmental agreements implementation and the emergence of cross-border issues. Our results illustrate current networks’ structure at the international level and at the same time at the continental level, suggesting the prospects of improving strategic partnerships for achieving sustainable development goals. Further research should explore exponential random graph models (ERGMs) by considering different specific attributes (i.e., GDP, country density, country population, types of landscapes) along with a detailed analysis, targeting how the principles of these multilateral agreements are implemented in practice at national level after ratification.

Keywords Cooperation networks · International environmental treaties · Clusters · Network structure · Brokers · Total link strength

A. Nita (✉) · L. Rozyłowicz
University of Bucharest, Center for Environmental Research, Bucharest, Romania
e-mail: andreea.nita@cc.unibuc.ro

1 Introduction

International cooperation contributes to improving the quality of environment and preserving natural resources for future generations, which became a priority of national policies over the last few decades. From this point of view, international environmental agreements have become a widely used legal solution to solve transboundary environmental problems that required urgent action and collaborative environmental governance [1]. The interest in such multinational environmental agreements represents the basis of a more cooperative world aiming to solve global environmental issues [2]. These approaches draw attention to the severity of cross-border environmental issues generated by country-level non-compliance with sustainable principles [3].

The rules established by environmental agreements contributed significantly to reducing several major environmental problems. Nevertheless, the high demand for resources and the economic development registered in the last two decades aggravated existing environmental issues such as: climate change, land-use change, species extinction, biodiversity crisis, excessive pollution, and massive deforestation [4]. In recent years, science advisers around the world advanced solutions for increasing collaboration between institutions and stakeholders at multiple scales, from local to regional, national and international [5–7], considering at the same time the social-ecological approach [8].

Given the large number of actors involved and conflicts that may arise in connection with environmental resources management, social network analysis can help find inspiring collaboration patterns [9]. Social network analysis represents a well-developed research field that uses network theory to statistically analyze the connections between stakeholders [10]. The analysis of the patterns of collaboration has been used in a wide range of disciplines, including the investigation of the complex stakeholders' networks involved in environmental conservation, research, and management [11–14].

At the international level, environmental agreements can be seen as arenas of collaboration between two or more parties, among them being established links, exchange of information, dissemination, or co-participation in activities with the same purpose [15]. From this perspective, environmental agreements should be considered as *communities* [16] with the same goal or as *networks* of stakeholders with closely same interests in solving environmental issues [3, 17, 18].

Analyzing the patterns in such complex collaboration network formed to promote environmental sustainability can help to identify the most influential actors [19], the findings illustrating some structural features that can offer advice for improvement and, respectively, ways to overcome existing barriers in effective environmental management and collaborative impact minimization [8]. However, the gap between the initial principles of the agreements and implementation is often discussed as one of the main causes of environmental crises that our society is currently experiencing [20].

Many important questions regarding the collaboration patterns established for solving environmental issues remain unexplored. A detailed structural analysis of the networks created around the states involved in promoting and implementing globally relevant environmental policies, such as United Nations landmark environmental policies, is needed. In this study, we use a two-mode network-based framework to analyze the structural patterns of the United Nations based cooperation network. The study identifies key promoters of international cooperation in environmental protection, collaborative countries, and leading environmental issues subject to international agreements globally and at a continental level.

Most of the studies carried out to date have placed a special emphasis on the structure of international environmental law and governance [2, 3] and tried to capture the basics of the connections created between its components [21]. In addition to these, our research attempts to fill the knowledge gap by revealing and investigating partnership patterns considering a structural continental perspective, which to our knowledge was not considered so far by using the network analysis approach.

This chapter represents an extension of our previous conference paper [15], in which we investigated the regional collaboration in the most popular and important international environmental treaties (considering the United Nations Treaty Database). In the present study, we analyze the structural patterns of the cooperation network to identify key promoters of worldwide cooperation, most collaborative countries, and most important environmental problems tackled by international agreements.

2 Concepts and Methodology

2.1 Network Data

This study focuses on the main environmental treaties listed under Collection Chapter XXVII: Environment (available online at <https://treaties.un.org>, accessed on 1st of August 2020). To perform the network analyses, we investigated the collaboration network created around the international environmental agreements presented in Table 1. For this, we extracted the name of the environmental treaty, signatory parties and introduced as attribute the continent for each node (actor), year of adoption, and main environmental issues tackled.

2.2 Research Design and Methods

Most collaboration networks are classified as one-mode [22], characterized by a single set of nodes of the same type, between which some connections are

Table 1 List of analyzed multilateral treaties (source: United Nations Treaty database)

| Id | Name of the multilateral environmental agreement | Place and date |
|----|--|-----------------------------------|
| 1 | Convention on Long-Range Transboundary Air Pollution | Geneva, 13th of November 1979 |
| 2 | Vienna Convention for the Protection of the Ozone Layer | Vienna, 22nd of March 1985 |
| 3 | Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and their Disposal | Basel, 22nd of March 1989 |
| 4 | Convention on Environmental Impact Assessment in a Transboundary Context | Espoo, 25th of February 1991 |
| 5 | Convention on the Protection and Use of Transboundary Watercourses and International Lakes | Helsinki, 17th of March 1992 |
| 6 | Convention on the Transboundary Effects of Industrial Accidents | Helsinki, 17th of March 1996 |
| 7 | United Nations Framework Convention on Climate Change | New York, 9th of May 1992 |
| 8 | Convention on Biological Diversity | Rio de Janeiro 5th of June 1992 |
| 9 | Agreement on the Conservation of Small Cetaceans of the Baltic North-East Atlantic, Irish and North Seas | New York, 17th of March 1992 |
| 10 | United Nations Convention to Combat Desertification in those Countries Experiencing Serious Drought and/or Desertification, Particularly in Africa | Paris, 14th of October 1994 |
| 11 | Lusaka Agreement on Cooperative Enforcement Operations Directed at Illegal Trade in Wild Fauna and Flora | Lusaka, 8th of September 1994 |
| 12 | Convention on the Law of the Non-Navigational Uses of International Watercourses | New York, 21st of May 1997 |
| 13 | Convention on Access to Information, Public Participation in Decision-Making and Access to Justice in Environmental Matters | Aarhus, 25th of June 1998 |
| 14 | Rotterdam Convention on the Prior Informed Consent Procedure for Certain Hazardous Chemicals and Pesticides in International Trade | Rotterdam, 10th of September 1998 |
| 15 | Stockholm Convention on Persistent Organic Pollutants | Stockholm, 22nd of May 2001 |
| 16 | Protocol on Civil Liability and Compensation for Damage Caused by the Transboundary Effects of Industrial Accidents on Transboundary Waters to the 1992 Convention on the Protection and Use of Transboundary Watercourses and International Lakes and to the 1992 Convention on the Transboundary Effects of Industrial Accidents | Kiev, 21st of May 2003 |
| 17 | Minamata Convention on Mercury | Kumamoto 10th of October 2013 |
| 18 | Regional Agreement on Access to Information, Public Participation and Justice in Environmental Matters in Latin America and the Caribbean | Escaz, 4th of March 2018 |

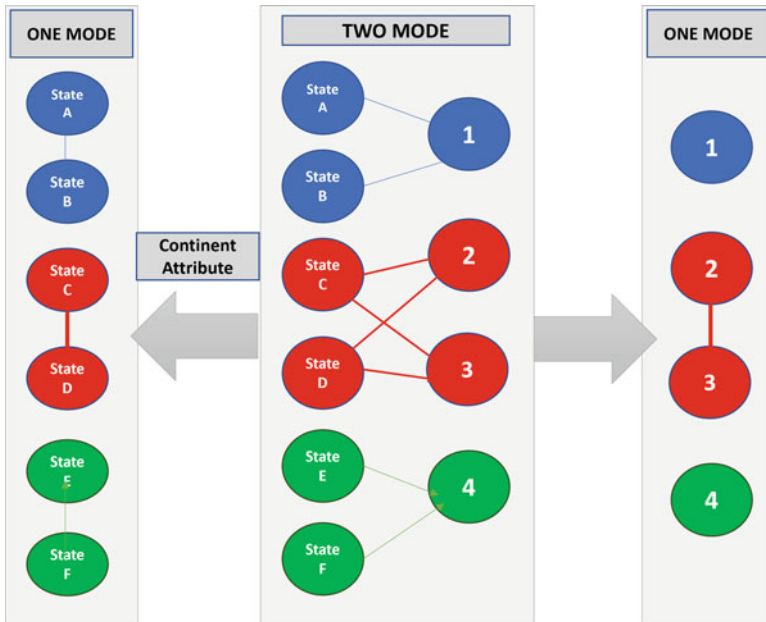


Fig. 1 Conversion of the state to treaty two-mode network into state to state and environmental agreement to environmental agreements one-mode networks; letters = participating parties to a treaty; numbers = different treaties

established (e.g., state to state collaboration). Other studies use networks with two different sets of nodes, where connections in the network are established only between nodes belonging to a different set (e.g., cooperation between states via treaties). Such networks are known as two-mode or bipartite networks [23, 24].

Using the environmental treaties listed in Table 1, we created a two-mode network [23, 25], containing as the first set of nodes the multilateral environmental agreements and the ratifying parties (i.e., collaborating states) as the second set of nodes (Fig. 1). To illustrate the level of involvement and interest, we first analyzed the patterns of the one-mode multilateral agreements network to highlight the interest between the common themes and countries, considering the total link strength. This indicator highlights the number of countries signing at least two agreements [26]. We further extracted the one-mode countries network to illustrate the country collaboration patterns clustered by considering their participation in similar agreements. To visualize the tendency of countries to cluster together [27], the same color frames group those particular countries in the same cluster, indicating that they usually participate in the same environmental agreements [26].

The next step in our analysis was to add the *continent* as an attribute for states. By extracting the continent-level graphs, we analyzed the centrality (degree centrality—showing the level of involvement of each state within most important environmental concerns and strategies) [28] and position of states in the network

(betweenness centrality) at the continental level [29]. We investigated these metrics to identify the intensities or strengths on the one hand and to pursue potential opportunities for information exchange [6]. Networks were represented using VOS mapping technique, considering the association strength normalization [30], where the position of the nodes in the collaboration network puts close together the strongly related nodes in clusters of the same color [30, 31]. To create the one-mode and two-mode matrices for analysis, we used UCINET software [25]. In conclusion, our study discusses and analyzes the networks composed of the most important players at the international level in environmental protection.

3 Research Findings

3.1 *One-Mode Analysis of Environmental Agreements and Participating Parties*

The one-mode network created using environmental agreements shows a close relation between eight agreements, namely between Vienna Convention for the Protection of the Ozone Layer (ID 2), Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and their Disposal (ID 3), United Nations Framework Convention on Climate Change (ID 7), Convention on Biological Diversity (ID 8), United Nations Convention to Combat Desertification in those Countries Experiencing Serious Drought and/or Desertification, Particularly in Africa (ID 10), Rotterdam Convention on the Prior Informed Consent Procedure for Certain Hazardous Chemicals and Pesticides in International Trade (ID 14), Stockholm Convention on Persistent Organic Pollutants (ID 15) and Minamata Convention on Mercury (ID 17). These agreements have a total link strength between 2780 and 3252, while the other agreements register scores between 162 (Lusaka Agreement on Cooperative Enforcement Operations Directed at Illegal Trade in Wild Fauna and Flora to a treaty ID 11) and 1240 (Convention on Long-Range Transboundary Air Pollution ID 1) (Fig. 2).

From the centrality perspective, the treaties with the highest degree and highest betweenness scores are the Vienna Convention for the Protection of the Ozone Layer (ID 2), followed by the United Nations Framework Convention on Climate Change (ID 7), the Convention on Biological Diversity (ID 8), and the United Nations Convention to Combat Desertification in those Countries Experiencing Serious Drought and/or Desertification (ID 10), particularly in Africa (see Table 1).

Figure 3 illustrates the one-mode collaboration network established between the parties, with a total number of links of 19,109 and a total link strength of the network of 286,074. Our results distinguish a close collaboration between states divided into two major clusters (i.e., parties participating in the same agreements), one in which are found mostly the European countries (green cluster) and the rest of the world (blue cluster). Countries such as Belgium, Denmark, Finland, France, Germany,

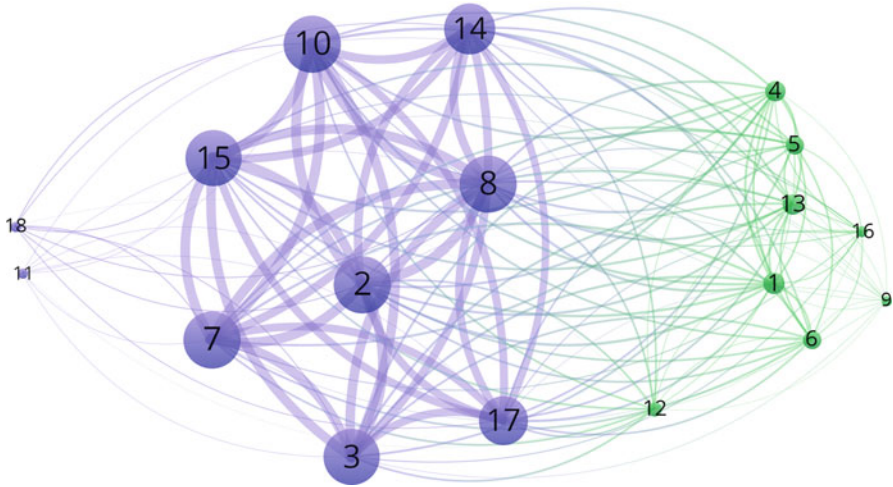


Fig. 2 Environmental agreements considering the number of signatory parties participating in two agreements (IDs of agreements in Table 1, size of nodes by total link strength)

Lithuania, Netherlands, Poland, Sweden, and the United Kingdom of Great Britain and Northern Ireland have the highest score for betweenness centrality (0.2), indicating that they have the best position in the network to promote international cooperation. Furthermore, Denmark, Finland, Sweden, and the United Kingdom of Great Britain and Northern Ireland have the highest scores for the eigenvector centrality, indicating that these are the most influential parties in the investigated network, holding a position that can influence the future from the perspective of implementing environmental agreements and restructuring the directions of interest in this field.

3.2 Two-Mode Analysis of Environmental Agreements and Participating Parties

The next section shows each party's involvement in the most important 18 environmental agreements at a continental level. Figures 4, 5, 6, 7, 8, and 9 present the networks for Asia—Fig. 4, Europe—Fig. 5, Africa—Fig. 6, North America—Fig. 7, South America—Fig. 8, and Oceania—Fig. 9. The environmental agreements presented in the lower left of each network represent the isolated treaties that did not have ratification from any of the continent's parties. For the Asian continent (Fig. 4), the network has 352 connections, grouped in 3 major clusters, countries such as: Qatar, Palestine, Iraq, Japan, Malaysia are more connected to the Convention on the Law of the Non-Navigational Uses of International Watercourses (ID 12), to the Rotterdam Convention on the Prior Informed Consent Procedure for Certain

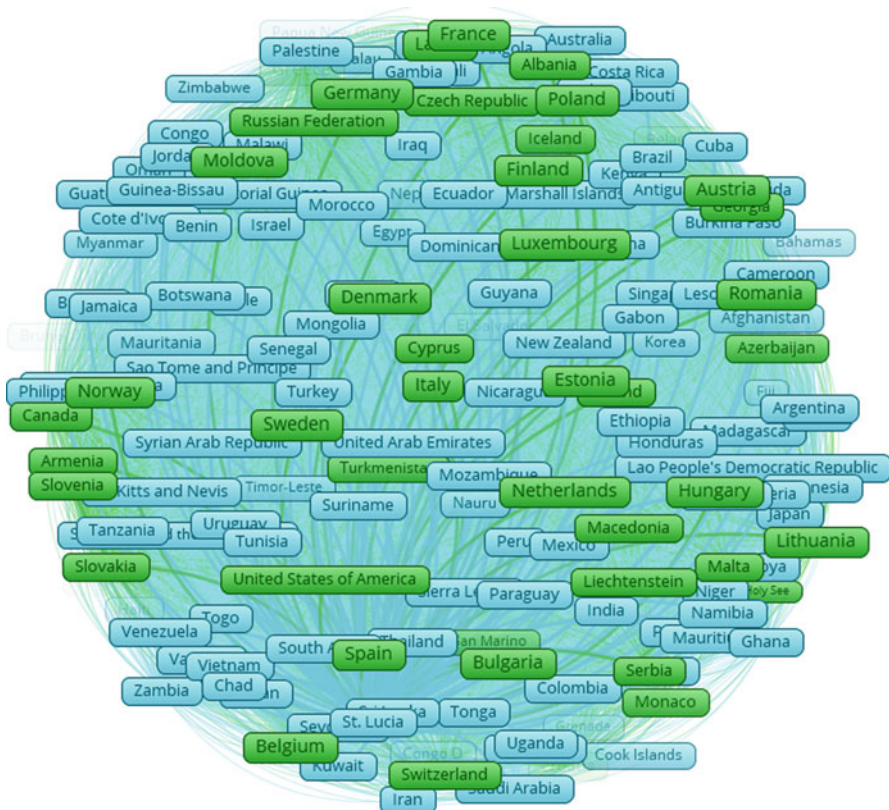


Fig. 3 Clustered country collaboration network for environmental sustainability

Hazardous Chemicals & Pesticides in International Trade (ID 14), to the Stockholm Convention on Persistent Organic Pollutants (ID 15), and to the Minamata Convention on Mercury (ID 17) (represented with green), Kazakhstan, Turkmenistan, and Uzbekistan are linked in the same cluster with the Convention on Long-Range Transboundary Air Pollution (ID 1), with the Convention on Environmental Impact Assessment in a Transboundary Context (ID 4), with the Convention on the Protection and Use of Transboundary Watercourses and International Lakes (ID 5), with the Convention on the Transboundary Effects of Industrial Accidents (ID 6) and with the Convention on Access to Information, Public Participation in Decision-Making & Access to Justice in Environmental Matters (ID 13), while India, Pakistan, Philippines are more related to the Vienna Convention for the Protection of the Ozone Layer (ID 2), to the Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and their Disposal (ID 3), to the United Nations Framework Convention on Climate Change (ID 7), to the Convention on Biological Diversity (ID 8) and to the United Nations Convention to Combat Desertification (ID 10) (see Table 1, Fig. 4).

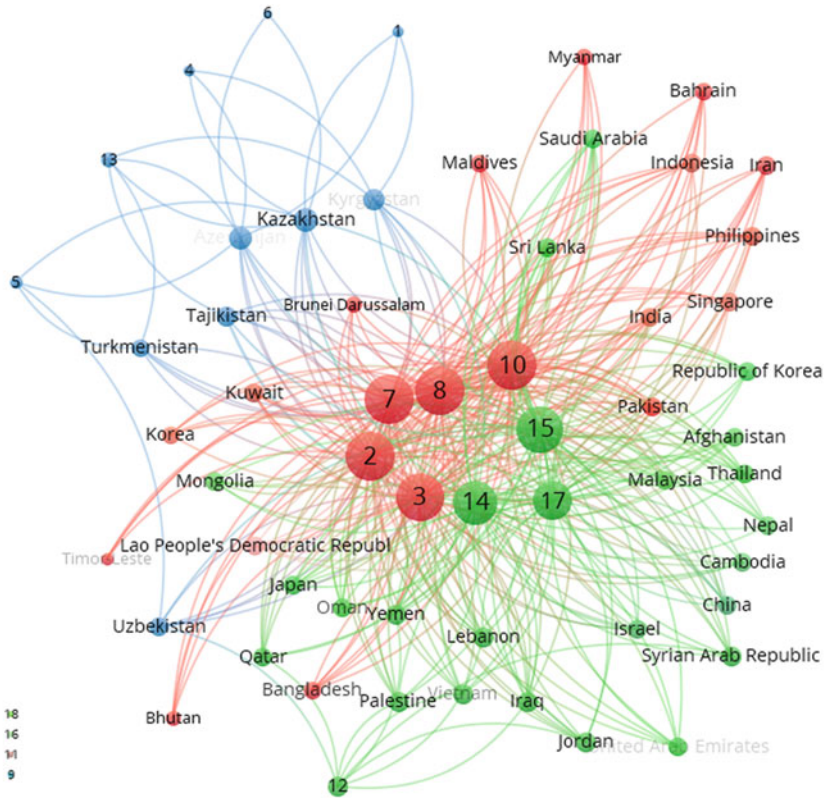


Fig. 4 Asia clustered two-mode network (treaties & ratifying parties, nodes size by link strength)

The network for the European continent is the most complex, with 632 links, having only two isolated agreements relevant for other world regions (Lusaka Agreement on Cooperative Enforcement Operations Directed at Illegal Trade in Wild Fauna and Flora (ID 11) and Regional Agreement on Access to Information, Public Participation and Justice in Environmental Matters in Latin America and the Caribbean, ID 18). Here, we observe 3 major clusters, fairly well distributed (Fig. 5). As the centrality results showed, this network has the most important and influential brokers of the entire international network (Fig. 3). The African network is presented in Fig. 6, having 444 links and 7 isolated treaties that are not ratified in this territory. The North American Continent patterns are similar to the other collaboration networks (see Fig. 7), having 190 links. The Oceania network has the highest number of isolated treaties (Fig. 9) and a low number of links (108), being very close to the South America network that gathers 107 links and only 10 ratified agreements out of total analyzed (Fig. 8).

In addition to the temporal changes that have marked the evolution of collaborations for environmental issues [15], the results of our analysis focused

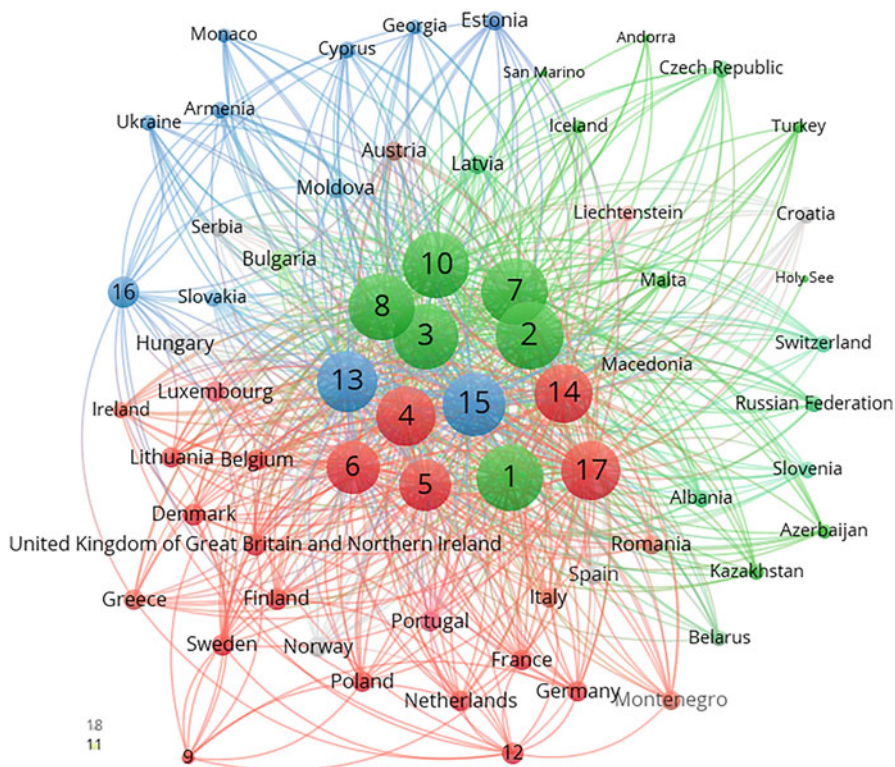


Fig. 5 Europe clustered two-mode network (treaties & ratifying parties, nodes size by link strength)

on the continental perspective (Figs. 4, 5, 6, 7, 8, and 9) could highlight the network degree of fragmentation in each continent. These results could contribute in solving disputes for common environmental resources, and increase collaboration for sustainable development.

4 Discussion and Future Work

The present research explored the patterns of collaboration between parties that ratified the most important 18 international environmental treaties implemented at the international level. The study helps to better identify the countries' interest in collaborating to solve and prevent the most pressing environmental problems. Our findings (Figs. 2 and 3) show significant progress regarding the interest in internationally addressing environmental issues. The basis of these partnerships and agreements are the networks of countries [18]. Our research results are relevant

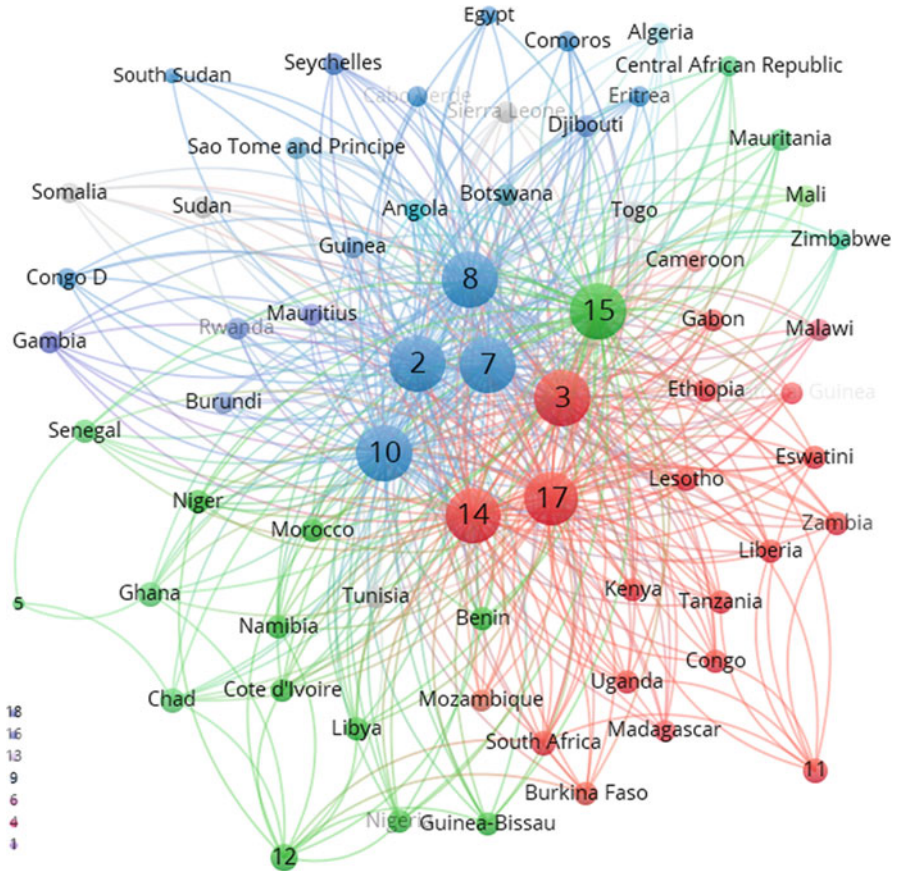


Fig. 6 Africa two-mode network (treaties & ratifying parties, nodes size by link strength)

for policymakers and other stakeholders interested in promoting global or regional environmental protection strategies [4].

As shown by the clustered models, one can observe a pattern of association of states in partnerships considering states' neighborhood, which is justified by neighboring countries' interest to solve and prevent cross-border environmental issues. At the same time, a greater centrality of some European states was noticed. The popularity of a state might be explained by the requirements to comply with other supranational rules, such as the European Union states that must comply with many environmental related EU Directives (e.g., EIA Directive 85/337/EEC or Directive 2001/42/EEC of the European Parliament and of the Council of 27 June 2001 on the assessment of the effects of certain plans and programs on the environment). At the continental level, a highly collaborative network of states and collaborative environmental governance [1, 14] is the key to ensuring optimal network flow and proper implementation of the environmental treaties

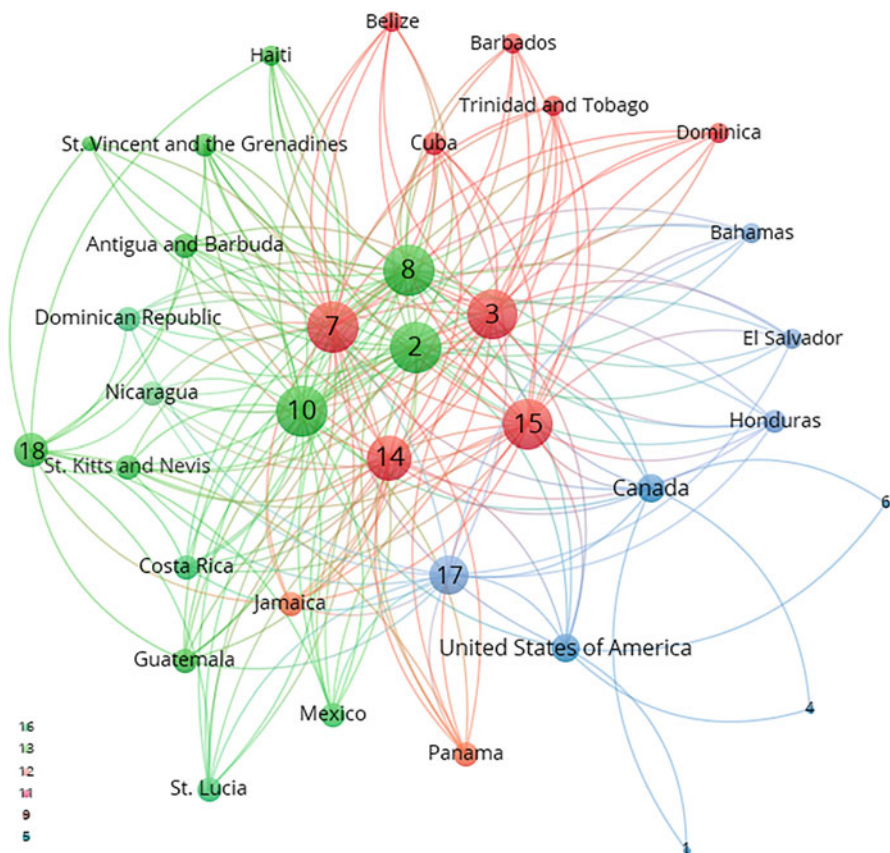


Fig. 7 North America two-mode network (treaties & ratifying parties, nodes size by link strength)

[32]. Considering that supranational environmental issues can be global, but also specific for each continent (Figs. 4, 5, 6, 7, 8, and 9), there is a need to stimulate the participation of all relevant stakeholders and the better implementation of the agreements' principles at the national level [10].

Furthermore, international environmental treaties are increasingly asking member states to include policy and societal dimensions into their work plans and strengthen their dissemination and science-policy interface activities. In this regard, better communication and transparency between practitioners, researchers, policy-makers, and other stakeholders have become a priority at the international level [15, 33]. Certainly, a much more complex and functional cooperation system is required to solve existing environmental conflicts [5, 34] and promote sustainable development from a strategic point of view [35]. A joint effort at the international level must be made to properly integrate environmental objectives into policy and practice [36]. Given that actors cooperate, compete, conflict, and support one another [5], these ties make a difference in finding the key to success [15, 37].

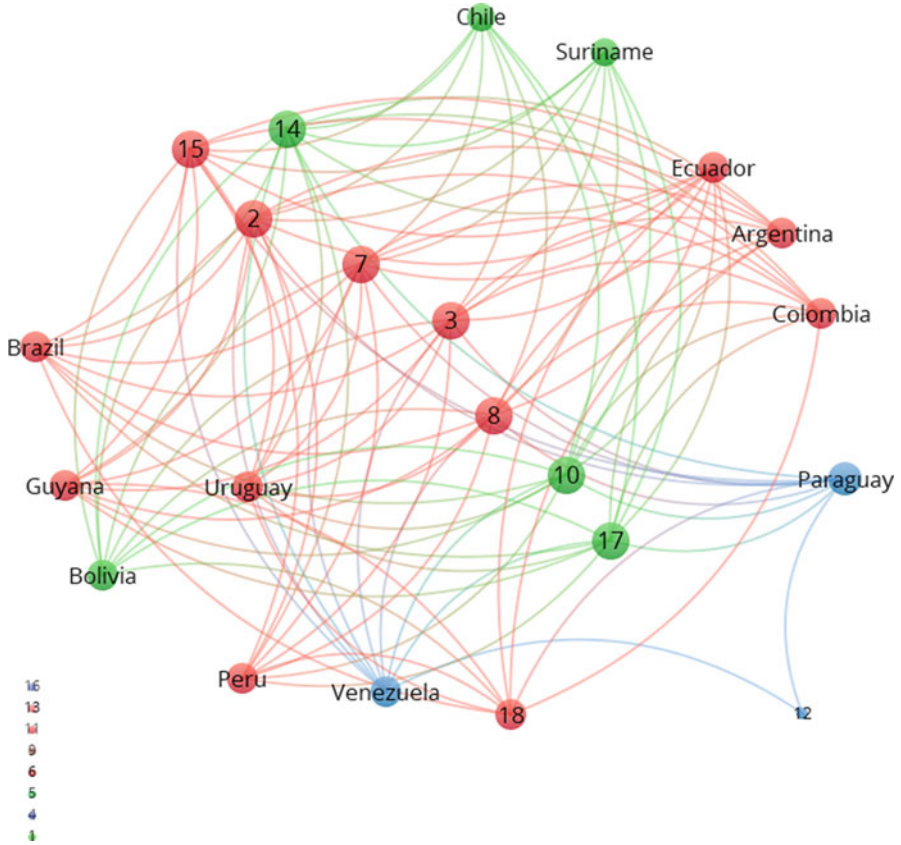


Fig. 8 South America two-mode network (treaties & ratifying parties, nodes size by link strength)

Consequently, the implicit benefits of using network analysis in investigating the international cooperation trend for solving transboundary environmental issues occur from the important highlights that reveal each actor's role within the network, along with drawing attention to the brokers that can positively influence. This information provides the necessary framework to be considered regarding the structure of the network. These findings contribute to the research field by offering the possibility to predict different scenarios [38] and to shape the 2050 long-term strategy for the environment [39].

The limitations of the present work could be the boundaries (only the most important environmental agreements) of our network; [6] however, the analyzed network covers all countries of the world, and so the results can be taken into account.

As future research topics related to the same topic discussed in this paper, a detailed analysis should also be developed on how the principles of these environmental agreements are put into practice at the national level. Furthermore,

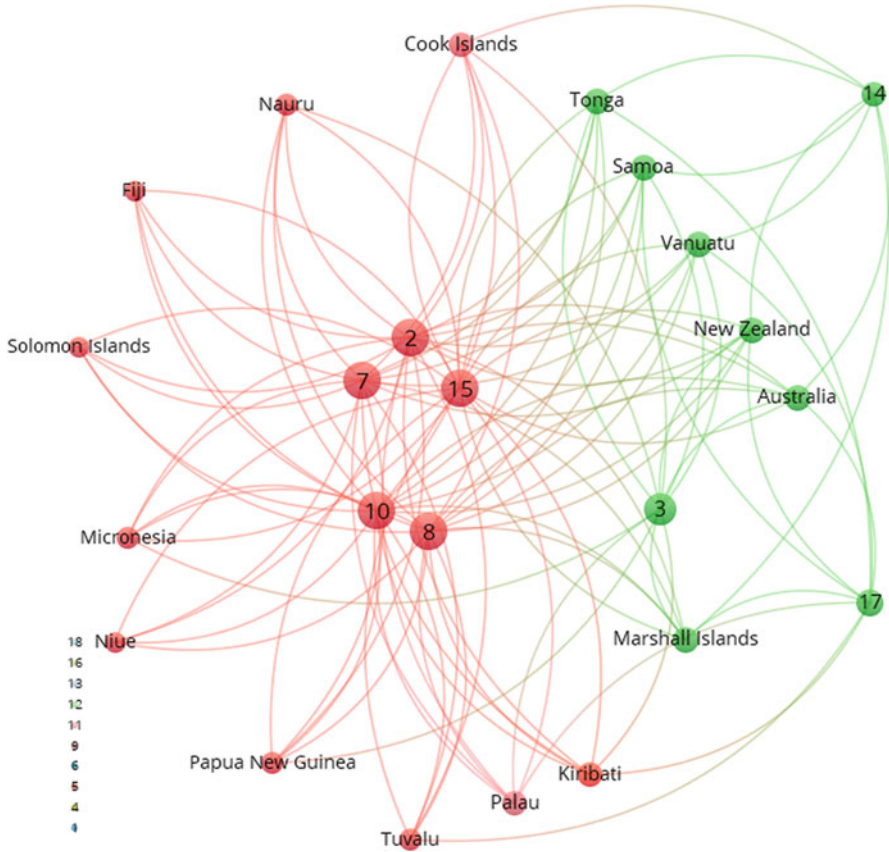


Fig. 9 Oceania two-mode network (treaties & ratifying parties, nodes size by link strength)

further studies should explore more complex network statistics such as exponential random graph models (ERGMs) [40, 41]. Such an approach may reveal the influence of node-level attributes (e.g., GDP, population density, landscape) on cooperation structures.

The network approach used in our investigation contributes to the research field by trying to analyze and integrate both social and ecological data, fundamentals in creating a coherent implementation of the environmental treaties and in changing management actions, so that ecological goals can be achieved without the emergence of social conflicts or environmental hazards [17]. From this perspective, our work tries to promote innovative concepts such as co-management and adaptive management [42].

Acknowledgments This research was supported by a grant of the Romanian National Authority for Scientific Research (<https://uefiscdi-direct.ro>), PN-III-P1-1.1-TE-2019-1039.

References

1. Bodin O (2017) Collaborative environmental governance: achieving collective action in social-ecological systems. *Science* 357(6352), art eaan1114. <https://doi.org/10.1126/science.aan1114>
2. Escobar-Pemberthy N, Ivanova M (2020) Implementation of multilateral environmental agreements: rationale and design of the environmental conventions index. *Sustainability* 12(17), art 7098. <https://doi.org/10.3390/su12177098>
3. Kim RE (2013) The emergent network structure of the multilateral environmental agreement system. *Glob Environ Chang* 23(5):980–991. <https://doi.org/10.1016/j.gloenvcha.2013.07.006>
4. Nita A (2019) Empowering impact assessments knowledge and international research collaboration – a bibliometric analysis of environmental impact assessment review journal. *Environ Impact Assess Rev* 78, art 106283. <https://doi.org/10.1016/j.eiar.2019.106283>
5. Hossu CA, Ioja IC, Susskind LE, Badiu DL, Hersperger AM (2018) Factors driving collaboration in natural resource conflict management: evidence from Romania. *Ambio* 47(7):816–830. <https://doi.org/10.1007/s13280-018-1016-0>
6. Berardo R, Fischer M, Hamilton M (2020) Collaborative governance and the challenges of network-based research. *Am Rev Public Adm* 50(8):898–913. <https://doi.org/10.1177/0275074020927792>
7. Manolache S, Nita A, Hartel T, Miu IV, Ciocanea CM, Rozyłowicz L (2020) Governance networks around grasslands with contrasting management history. *J Environ Manag* 273, art 111152. <https://doi.org/10.1016/j.jenvman.2020.111152>
8. Bodin O, Alexander SM, Baggio J, Barnes ML, Berardo R, Cumming GS, Dee LE, Fischer AP, Fischer M, Mancilla Garcia M, Guerrero AM, Hileman J, Ingold K, Matous P, Morrison TH, Nohrstedt D, Pittman J, Robins G, Sayles JS (2019) Improving network approaches to the study of complex social–ecological interdependencies. *Nat Sustain* 2(7):551–559. <https://doi.org/10.1038/s41893-019-0308-0>
9. Bodin O, Robins G, McAllister RRJ, Guerrero AM, Crona B, Tengö M, Lubell M (2016) Theorizing benefits and constraints in collaborative environmental governance: a transdisciplinary social-ecological network approach for empirical investigations. *Ecol Soc* 21(1), art 40. <https://doi.org/10.5751/es-08368-210140>
10. Bodin O, Crona B, Thyresson M, Golz AL, Tengo M (2014) Conservation success as a function of good alignment of social and ecological structures and processes. *Conserv Biol* 28(5):1371–1379. <https://doi.org/10.1111/cobi.12306>
11. Reed MS, Graves A, Dandy N, Posthumus H, Hubacek K, Morris J, Prell C, Quinn CH, Stringer LC (2009) Who’s in and why? A typology of stakeholder analysis methods for natural resource management. *J Environ Manag* 90(5):1933–49. <https://doi.org/10.1016/j.jenvman.2009.01.001>
12. Berardo R, Scholz JT (2010) Self-organizing policy networks: risk, partner selection, and cooperation in estuaries. *Am J Polit Sci* 54(3):632–649. <https://doi.org/10.1111/j.1540-5907.2010.00451.x>
13. Keskitalo EC, Baird J, Laszlo Ambjörnsson E, Plummer R (2014) Social network analysis of multi-level linkages: a Swedish case study on northern forest-based sectors. *Ambio* 43(6):745–58. <https://doi.org/10.1007/s13280-014-0492-0>
14. Nita A, Manolache S, Ciocanea CM, Rozyłowicz L (2019) Real-world application of ego-network analysis to evaluate environmental management structures. In: Karampelas P, Kawash J, Ozyer T (eds) From security to community detection in social networking platforms. *ASONAM 2017*, chap. 1. Lecture notes in social networks. Springer, Cham. https://doi.org/10.1007/978-3-030-11286-8_1
15. Nita A, Rozyłowicz L (2020) Dynamics of the international environmental treaties–perspectives for future cooperation. In: 2020 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp. 549–556. <https://doi.org/10.1109/ASONAM49781.2020.9381333>
16. Onnela JP, Arbesman S, Gonzalez MC, Barabasi AL, Christakis NA (2011) Geographic constraints on social network groups. *PLoS One* 6(4), art e16939. <https://doi.org/10.1371/journal.pone.0016939>

17. Ioja IC, Nita MR, Hossu CA (2016) Environmental conflicts. In: Paulo N, Davide C (eds.) *Interdisciplinary perspectives on contemporary conflict resolution*, chap. 4. *Advances in linguistics and communication studies*. IGI Global, Hershey, pp 56–79. <https://doi.org/10.4018/978-1-5225-0245-6.ch004>
18. Podolny JM, Page KL (1998) Network forms of organization. *Annu Rev Sociol* 24(1):57–76. <https://doi.org/10.1146/annurev.soc.24.1.57>
19. Song AM, Temby O, Kim D, Hickey GM (2020) Assessing the influence of international environmental treaty secretariats using a relational network approach. *Earth Syst Gov* 5, art 100076. <https://doi.org/10.1016/j.esg.2020.100076>
20. Manolache S, Nita A, Ciocanea CM, Popescu VD, Rozyłowicz L (2018) Power, influence and structure in natura 2000 governance networks. A comparative analysis of two protected areas in Romania. *J Environ Manag* 212:54–64. <https://doi.org/10.1016/j.jenvman.2018.01.076>
21. Newman ME (2004) Fast algorithm for detecting community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 69(6), art 066133. <https://doi.org/10.1103/PhysRevE.69.066133>
22. Lindelauf R, Borm P, Hamers H (2012) One-mode projection analysis and design of covert affiliation networks. *Soc Netw* 34(4):614–622. <https://doi.org/10.1016/j.socnet.2012.07.001>
23. Borgatti SP, Everett MG (1997) Network analysis of 2-mode data. *Soc Netw* 19(3):243–269. [https://doi.org/10.1016/s0378-8733\(96\)00301-2](https://doi.org/10.1016/s0378-8733(96)00301-2)
24. Opsahl T (2013) Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Soc Netw* 35(2):159–167. <https://doi.org/10.1016/j.socnet.2011.07.001>
25. Borgatti SP, Everett MG, Freeman LC (2002) UCINET for windows: software for social network analysis. Analytic Technologies, Harvard
26. Guo YM, Huang ZL, Guo J, Li H, Guo XR, Nkeli MJ (2019) Bibliometric analysis on smart cities research. *Sustainability* 11(13), art 3606. <https://doi.org/10.3390/su11133606>
27. Opsahl T, Panzarasa P (2009) Clustering in weighted networks. *Soc Netw* 31(2):155–163. <https://doi.org/10.1016/j.socnet.2009.02.002>
28. Nita A, Rozyłowicz L, Manolache S, Ciocanea CM, Miu IV, Popescu VD (2016) Collaboration networks in applied conservation projects across Europe. *PLoS One* 11(10), art e0164503. <https://doi.org/10.1371/journal.pone.0164503>
29. Abbasi A, Hossain L, Leydesdorff L (2012) Betweenness centrality as a driver of preferential attachment in the evolution of research collaboration networks. *J Informet* 6(3):403–412. <https://doi.org/10.1016/j.joi.2012.01.002>
30. van Eck NJ, Waltman L (2014) Visualizing bibliometric networks. In: Ding Y, Rousseau R, Wolfram D (eds) *Measuring scholarly impact*, chap. 13. Springer, Cham, pp 285–320. https://doi.org/10.1007/978-3-319-10377-8_13
31. van Eck NJ, Waltman L (2010) Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics* 84(2):523–538. <https://doi.org/10.1007/s11192-009-0146-3>
32. Csermely P, London A, Wu LY, Uzzi B (2013) Structure and dynamics of core/periphery networks. *J Complex Networks* 1(2):93–123. <https://doi.org/10.1093/comnet/cnt016>
33. Nita A, Ciocanea CM, Manolache S, Rozyłowicz L (2018) A network approach for understanding opportunities and barriers to effective public participation in the management of protected areas. *Soc Netw Anal Min* 8(1), art 31. <https://doi.org/10.1007/s13278-018-0509-y>
34. Gavrilidis AA, Nita A, Niculae MI (2020) Assessing the potential conflict occurrence due to metropolitan transportation planning: a proposed quantitative approach. *Sustainability* 12(2), art 527. <https://doi.org/10.3390/su12020527>
35. Hersperger AM, Bürgi M, Wende W, Bacău S, Grădinaru SR (2020) Does landscape play a role in strategic spatial planning of European urban regions? *Landsc Urban Plan* 194, art 103702
36. Runhaar H (2016) Tools for integrating environmental objectives into policy and practice: what works where? *Environ Impact Assess Rev* 59:1–9. <https://doi.org/10.1016/j.eiar.2016.03.003>
37. Barabási AL (2018) *The formula: the science behind why people succeed or fail*. Macmillan, London

38. Zhang S, Xu W, Wang K, Feng W, Athienitis A, Hua G, Okumiya M, Yoon G, Cho DW, Iyer-Raniga U, Mazria E, Lyu Y (2020) Scenarios of energy reduction potential of zero energy building promotion in the Asia-pacific region to year 2050. *Energy* 213, art 118792. <https://doi.org/10.1016/j.energy.2020.118792>
39. van Sluisveld MAE, Hof AF, van Vuuren DP, Boot P, Criqui P, Matthes FC, Notenboom J, Pedersen SL, Pfluger B, Watson J (2017) Low-carbon strategies towards 2050: comparing ex-ante policy evaluation studies and national planning processes in Europe. *Environ Sci Pol* 78:89–96. <https://doi.org/10.1016/j.envsci.2017.08.022>
40. Goodreau SM (2007) Advances in exponential random graph (p*) models applied to a large social network. *Soc Netw* 29(2):231–248. <https://doi.org/10.1016/j.socnet.2006.08.001>
41. Snijders TAB, Pattison PE, Robins GL, Handcock MS (2016) New specifications for exponential random graph models. *Sociol Methodol* 36(1):99–153. <https://doi.org/10.1111/j.1467-9531.2006.00176.x>
42. Kovács E, Mile O, Fabók V, Margóczy K, Kalóczkai A, Kasza V, Nagyné Grecs A, Bankovics A, Mihók B (2021) Fostering adaptive co-management with stakeholder participation in the surroundings of soda pans in Kiskunság, Hungary – an assessment. *Land Use Policy* 100, art 104894. <https://doi.org/10.1016/j.landusepol.2020.104894>

Critical Mass and Data Integrity Diagnostics of a Twitter Social Contagion Monitor



Amruta Deshpande, Vladimir Barash, Clayton Fink, Christopher Cameron,
Aurora Schmidt, Wei Dong, Michael Macy, and John Kelly

Abstract We expand on our previous efforts to describe and validate a system for monitoring social contagions on Twitter [2]: social movements, rumors and emotional outbursts that spread from person to person in a viral manner. In this effort, which is an extended rendition of [1], we additionally describe added movement tracers (@mentions and URLs) to a productionalized version of the Contagion MonitorTM. These allow for different tracers to capture movements and information that spread. We additionally include new analyses to confirm critical mass behavior on Twitter and to characterize data limits. A final addition in this extended effort is the merging of metrics to distinguish organic human activity from the coordinated. We draw from a parallel effort to pull metrics to measure coordination and leverage them here. The Contagion MonitorTM parses Twitter posts to identify emerging phenomena, as captured in hashtags, URLs, words and phrases, or account-handles, and then determines the extent to which a particular phenomenon spreads via the social network (in contrast to its spread via news broadcasts or independent adoption), by approximating its adoption threshold. It makes a second judgment about whether the phenomenon has reached critical mass. We tested [2] our prototype monitor on two data sources—an ongoing stream of tweets grouped by user-added tags and a collection of posts by a monitored set of Nigerian Twitter users—before productionalizing. We used the Amazon Mechanical Turk platform to evaluate the performance on both data sources. In both cases, we found that our approach successfully distinguished between high-threshold and low-threshold social contagions. Additionally, we now confirm critical mass behavior for

A. Deshpande (✉) · V. Barash · J. Kelly
Graphika Labs, Graphika Inc., New York, NY, USA
e-mail: amruta.deshpande@graphika.com; vlad.barash@graphika.com; john.kelly@graphika.com

C. Fink · A. Schmidt
Applied Physics Laboratory, Johns Hopkins University Maryland, Baltimore, MD, USA
e-mail: aurora.schmidt@jhuapl.edu

C. Cameron · W. Dong · M. Macy
Social Dynamics Lab, Cornell University, Ithaca, NY, USA
e-mail: cjc73@cornell.edu; wd96@cornell.edu; mwm14@cornell.edu

the prototype. For the productionalized version, we further diagnose the integrity of our data collection process and describe the newly incorporated tracers of infectious Twitter messages. Finally, we determine if the spread of a contagion is coordinated, enabling the tool to surface coordinated activity and distinguish it from organic grassroots movements. This addition provides key context around transformative movements in social media.

Keywords Social Network Analysis · Critical Mass · Social Contagion · Complex Contagion · Networks · Twitter · Nigeria · Coordinated Social Movements · Inorganic Online Social Activity · Coordination Framework

1 Introduction

In this work, we extend our efforts presented in [2]. We re-include aspects of the prior work for continuity, and then describe additional new efforts that build on the prior work. In this way, we present here a coherent and expanded effort on our sophisticated analytical system and tool—the Contagion Monitor™.

This past decade has seen transformative social movements, such as the Arab Spring, Black Lives Matter, and the movements which led to elections of Barack Obama and Donald Trump to the US presidency. Recent approaches in social science [1, 6, 7, 10, 21, 24] seek to understand the dynamics by which social movements break out of local contexts to become widespread phenomena. We leverage these approaches to construct an automated tool, the Contagion Monitor™, to detect emerging social movements before they gain widespread popularity. Our tool expressly seeks out the more fundamental and transformative social movements in social media, which are capable of bringing about real behavior change. In so doing, the tool offers an advantage over other approaches (see Sect. 2.3) which look for trends and perform general social listening tasks. We test this tool in two large-scale empirical settings and have subsequently deployed it in a productionalized application that is currently in daily use.

Our innovation relies on a network structural approach that can distinguish between transient movements (e.g., viral memes) of low social impact, and more transformative ones (e.g., voting or protesting). We are able to make this distinction based on how information cascades move through social networks, without relying on time-consuming language analysis or contextual searches. Our approach also focuses on anticipating cascades, and thus would enable analyst end users to monitor social contagions before they become widely popular.

We define “social contagions” as social movements, rumors, and emotional outbursts that spread from person to person. Our social Contagion Monitor processes streams of Twitter data to scan for these. For ethical reasons, we avoid running any sort of controlled experiment on large human populations; instead, we treat data collected from the Monitor as purely observational or as “natural experiment” outcomes.

In scanning for social contagions, the Monitor makes two key algorithmic judgments about their dynamics. For one, the tool calculates metrics to estimate the threshold to participate in the movement[4], indicating its transformative potential; and for the other, it looks to see whether the movement is about to break out of its local cluster[1], indicating virality. Together, these judgments can help researchers differentiate between three categories of movements (from the transient to beyond):

- (a) Low-threshold movements that have low cost to participation or network externality: these can spread quickly, but some scholars (e.g., [4]) speculate they are unlikely to bring about transformative political or social change.
- (b) High-threshold movements that have not yet broken out of local network clusters: these movements may be important to monitor but do not yet have the reach to bring about transformative political or social change.
- (c) High-threshold movements that have broken out of local networks: these are both costly (i.e. sufficiently “high stakes”) and have the reach to have a relatively high probability of affecting political or social change.

We test two setups of the Monitor: the regional Contagion Monitor (RCM) and the streaming Contagion Monitor (SCM). These implementations differ by the type of data they consume and the criteria for detecting emerging phenomena. The RCM focuses on a smaller geographic region and employs significant historical data for detecting new contagions while the SCM processes streaming data with minimal use of historical data.

With the RCM, we test our system in a more controlled setting of a fixed, more narrowly scoped community of regional Twitter users over long time. Here, we can determine which diffusion events—captured via hashtags, URLs, or words and phrases—are propagating according to theoretical expectations. If productionalized, this setup would offer regionally focused insights into emerging movements and their transformative potentials. With the SCM, we extend the scope of the monitor to a wider range of sociocultural settings, dynamically capturing the engaged network and allowing the setup to pick out more recent, potentially transformative worldwide movements. The trade-off between these setups is one of greater depth (RCM) versus breadth (SCM) of topics and regions. We have currently productionalized the SCM setup for daily use.

In this work, we continue a body of testing and validation of our two key judgements, recapturing and extending our efforts in [2]. A detailed analysis of novel metrics in the RCM setting, in the context of hashtags used by a local Nigerian community, is described in [7]. We validate the threshold based judgement in both monitor settings, and describe the SCM set up in [2] (repeating here for continuity). In this work, we newly report on the validation of the second judgement our Monitor makes, about the network reach of contagions that break out of a local subnetwork. Furthermore, we conduct tests to assess data needs for reliable measurements of our metrics in a productionalized setting first introduced in [2]. Finally, we describe an integration of our Monitor with a framework [9] for identifying coordinated behavior to help improve the quality of our Monitor. This

integration enables our Monitor to distinguish between organically viral movement tracers and coordinated, inorganically grown movements; and thus incorporate one of the future improvements we identified in [2].

This paper is structured in the following way. We begin by describing prior theoretical work and other social media trend tools in the next section. In the following section (Sect. 3), we describe the monitoring setups, including what data they ingest and some risks. In Sect. 4, we describe our evaluation framework for the monitors' performance, as embedded within some research questions, and share those results. Finally, we conclude in Sect. 5 with a discussion of results and some future work.

2 Background

2.1 Social Movements and Contagion Models

Reference [19] formulated a model of threshold-based adoption cascades on populations in lattice networks, where the threshold depends on k , the number of network neighbors of a node. Reference [4] extended [19]'s model to the Small World [25] network model, which involves the random rewiring of ties on lattice networks; a closer approximation to real social networks. Reference [25] tested the spread of simulated simple and complex contagions on Small World networks. Simple contagions, defined by their threshold equal to $1/k$, model the spread of disease, information, and easily adopted behaviors that do not require a confirmatory (and redundant) exposure to become infected. In contrast, complex contagions, defined to have thresholds greater than $1/k$, model behaviors like protesting, rioting, or adoption of new conventions or technologies that carry a greater cost to adoption and require more than one confirmatory exposure, prior to infection.

Reference [4] found that in Small World networks, complex contagions rarely spread beyond the initial "seed" cluster, except occasionally when these can leverage the shortcuts that result in Small World networks from the random re-wiring of ties to new clusters. Typically, the re-wiring reduces the confirmatory exposures available to the complex contagion to spread. However, reference [1] discovered a *critical mass* point, in the fraction of infected nodes, beyond which the contagion spreads through the full network quickly and with high probability (like simple contagions). This happens, because beyond the critical mass, the chance of sufficient exposures to leverage shortcuts to new clusters becomes very high.

The authors in [1] confirmed through simulation that complex contagions require a dense seed cluster, in the initial stages of propagation, to reach critical mass. They found that the critical mass point has a two-part statistical signature: (1) the contagion's rate of propagation dramatically changes from negative (as the contagion begins to saturate the local seed cluster) to positive (as the contagion begins to spread in fresh network regions) and (2) there is a sharp drop in the density

of the network neighbors of new adopters. The first indicates that the contagion has broken out of the seed cluster while the second indicates that it is growing in an unsaturated region. Together, these mean that the contagion is leveraging the randomly re-wired long-range ties and is not limited by the local structure where it begins.

2.2 Social Movements and Social Media

Many researchers have attempted to study viral behavior in general, and complex and simple contagions in particular, in online social network data. Work by Romero et al. [21] observed that politically-themed hashtags behave like complex contagions, whereas hashtags corresponding to neologisms and Twitter idioms behave like simple contagions. Reference [10] found evidence of social influence and complex contagions in Twitter recruitment networks around social mobilization in Spain in May 2011. Reference [20] compiled a large overview of viral messages, including those related to social contagions, and pointed to the important role for “gatekeepers” whose influence can cause a social contagion to go viral. Reference [7] found the hashtag #bringbackourgirls, relating to the movement to bring back hundreds of girls kidnapped by Boko Haram in Nigeria in 2014, resembled a complex contagion.

Reference [24] found that the diffusion of campaign donations is a complex contagion driven by independent social reinforcement. The authors found that people are more likely to donate if exposed to donors from different social groups than equally many donors from the same group, which suggests an important extension of the complex contagion model: high threshold contagions may require not just multiple sources of exposure but multiple independent sources. Accordingly, we equipped our Contagion Monitor to be able to identify the distribution of adopters across social groups.

2.3 Social Media Monitoring Tools

There are a number of social media monitoring tools, both in academic [16, 23] and industrial [11, 15, 22] settings. Our monitor offers some advantages over these. The Observatory on Social Media and NIFTY analyze the spread of information rather than of social contagions specifically. Crimson Hexagon and other industrial tools focus on general monitoring (influencer identification, trends, geo-location) rather than the specific identification of social contagions and analysis of their dynamics as we do. Our metrics, by design, identify high-threshold social contagions, more likely to be of actionable interest. By analyzing the dynamics of information cascades, and using simple related metrics, we offer a faster and independent signal

of emerging movements. Our Monitor is thus compatible with and goes beyond the capabilities of existing tools.

3 Methods and Data

In this section, we describe the regional, streaming, and productionalized Contagion MonitorTM settings, with respect to the data they ingest, the similarities and differences in their processes, and describe their outputs. All monitors undergo the same set of steps:

- (a) candidate selection
- (b) data collection per candidate
- (c) network collection per candidate, and
- (d) contagion analysis.

Before describing the data ingested by these steps, we address some risks associated with our social Contagion Monitoring tool, related Institutional Review Board (IRB) considerations, and our approach to their mitigation. In Sect. 3.7, we briefly describe the data, methods, and integration of the Coordination Framework [9] which we leverage to flag coordination in spreading movements. Evaluations of the methods and results follow in Sect. 4.

3.1 IRB Compliance

Our research received an IRB exemption because we analyze publicly available data that we also de-identify. We have made it a priority to minimize the risk of exposing sensitive personal data, whether through a leak or hack of our data store, or through unwitting exposure via publication. As a result, we retained in the productionalized tool, the de-identification steps we applied to network collection in the RCM and SCM.

3.2 Risks of Collecting Data on Social Movements

Our data collection is associated with two major sources of risk. First, the data we collect are public Twitter posts, which may contain personally identifying information. A Pew internet survey of teenage use of social media [18] shows that 24% of teens used Twitter in 2012. The same survey shows that 91% of teens post a photo of themselves on Facebook, 71% post their school name, and so on, in the same time period. The scale of our data collection is highly likely to result in

including personally identifying information on minors, which requires special care to store or process [13].

Our streaming and regional Contagion Monitors analyze social contagions, which include social movements. The second risk associated with our data collection is that some of these movements (like the pro-LGBT movement in Nigeria) carry severe risks for participation, including imprisonment. The Same Sex Marriage Prohibition Act in Nigeria “imposes a 10-year prison sentence on anyone who “registers, operates or participates in gay clubs, societies and organization” or “supports” the activities of such organizations” [14]. Simply exposing the identities of Nigerian individuals involved in a pro-LGBT hashtag on Twitter could subject such individuals to the effect of this law. We recognize the severe risks of improper data management when constructing the social Contagion Monitor.

We do not publish tweets from our Contagion Monitors except as one-off examples, in which case, we blur all identifying information in the tweet text or metadata. In all publications (including this one), we report summary statistics about hashtags and other features identified by the social Contagion Monitor rather than individual-level behavior.

3.3 *Data Ingestion*

3.3.1 **Streaming Contagion Monitor**

The SCM collects streaming, public Twitter posts (or tweets) to perform its contagion analysis and contagion candidate selection. In this study, in order to evaluate complex contagion theory from our results, we scope our collection on themes associated with thirty of [11]’s library of social media network maps. These themes span multiple sociocultural contexts—US and European politics, industry verticals such as automotive and food, and large sports events. Reference [11] constructs themed maps, or networks of Twitter accounts that engage with conversations on a theme, and then collects the accounts’ live streaming tweet activity.¹ We access these data streams to perform candidate selection. Importantly, we note that these data are not filtered geographically or by keyword, and truly represent the conversation at large on Twitter. We deliberately chose both political and non-political maps to capture both high-threshold social movements and low-threshold news events.

The SCM passes the streaming data to our sliding window monitor (SWM), which is capable of real-time tweet stream processing. It does 3 main tasks: feature extraction, tracking features, and nominating features of interest. The SWM acts as a front line filter, reducing the volume of data that is passed on to later analysis stages

¹ `statuses/user_timeline` endpoint—https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.

(including candidate selection). In this study, the features are based on hashtags alone, however, the SCM and SWM are capable of additionally processing Twitter account-handles, account-mentions, URLs, words, and bi-grams.

For each incoming tweet, the SWM extracts the hashtag data and creates features that represent hashtags, retweeted hashtags, hashtag pairs and retweeted hashtag pairs. Hashtag pairs are sets of hashtags that appear together in the same tweet. The SWM tracks the frequency of each feature over sliding intervals of 10 min, 1 h and 4 h, reporting any features that exceed a threshold number of appearances. The SWM aggregates reports on these to identify the most popular features over the last 24 h and nominates the top features for further analysis.

3.3.2 Regional Contagion Monitor

We invoke the regional Contagion Monitor, built and studied in [8], as a static network on which to compute contagion measures for comparison against the same measures computed for the SCM. The RCM focused its data collection on networked accounts that were likely to be located in a specific geographic region. In this way, we approximate a physical social network that is static, and measure its social media activity to look for contagions.

Full details of the RCM data collection are available in [8], but we summarize some key elements here. We identified a set of 108,744 Twitter users located in Nigeria² between January and mid-April of 2017, that also remained public and active through our later data collection, and for whom we could carefully assess location. We constructed a network of these users by leveraging the “follows” relationship on the platform, and collected approximately 270 million tweets³ between April and November of 2017. Our final graph, on which we computed contagion measures, consisted of 103,659 nodes, 11,753,606 edges and an average degree of 113.

3.3.3 Productionalized Streaming Contagion Monitor

The productionalized Contagion MonitorTM setup, which is in use today, is based on the SCM, but is further adapted to [11]’s pre-parsed data stores and collection. It thus bypasses the SWM’s input handling, but computes the same metrics for candidate selection from high resolution feature (e.g., hashtags, URLs) occurrence time-streams computed from the [11] data. The productionalized tool currently processes Twitter hashtags, account-mentions, and URLs as contagion candidates. The tool additionally implements an optional, alternate logic for candidate selection, based on slightly different criteria that are described more in the next section. The

² <https://www.cia.gov/library/publications/the-world-factbook/geos/ni.htm>.

³ `statuses/user_timeline` public Twitter Application Programming Interface (API) endpoint.

real-time tweet-stream processing capability, described for the SCM, was retired for this productionalized tool due to the greater demand on resources (processing time) and the data collection limitations associated with Twitter collection (rate limits) that relate to the later analysis steps.

3.4 Candidate Hashtag Selection

3.4.1 Streaming Contagion Monitor

The candidate selector collects the most popular features over the previous 24h, and then applies a multi-step filter and an exclusion-list filter. The multi-step seeks to filter out features that have appeared in the top 300 most popular, at any time in the past 5 days, with one exception. If in the last 5 days, the number of communities identified by Graphika [11] in which the feature is relevant [17] has increased by 20%, then it is not filtered out. The exclusion-list filter removes hashtags that are spam or regularly reoccurring Twitter “memes” (e.g. #followfriday) that are obviously not related to social movements.

3.4.2 Regional Contagion Monitor

For each calendar day of tweets (assuming midnight as GMT+1 or West African Time), the regional Contagion Monitor extracted all hashtags used during that day and the 30 days prior to that. For those hashtags used by 100 or more unique users, we classified a hashtag as a candidate for analysis if its count for that day was two standard deviations greater than its mean count for the previous 30 days. Our contagion analysis for a given day was restricted to these hashtags. We excluded from analysis hashtags that were on our exclusion-list (common ones such as #NP) or hashtags used by monitored users in the six months prior to August 1, 2017. The monitor nominated 2,823 trending hashtags from August through October 2017.

3.4.3 Productionalized Contagion MonitorTM

The productionalized tool was developed with two possible modes for candidate selection, the *general* and the *dedicated*. These two differ in one main way. In the general mode, the tool undergoes the same candidate selection process as the SCM, choosing from among the top 300 most popular, and applying the subsequent multi-step and exclusion-list filters.

In the dedicated mode, it additionally accepts a list of maps as input, and makes an attempt to estimate which of the top 3000 most popular features from the previous day in [11]’s data stores are most relevant to the list of input maps. It then also applies the same multi-step and exclusion-list filtering as the SCM. This dedicated

mode, allows an end-user to promote a more relevant or interesting set of features, based on theme or topic, that is less biased by features that are popular on Twitter.

3.5 Contagion Analysis

3.5.1 Streaming Contagion Monitor

For each feature identified by the candidate selector, the contagion analyzer collects up to the last five days of tweets that have used (adopted) the feature. From the set of adopters identified in this collection, it does a second round data collection to collect and construct a network for each adopter [7]. The edges in the network connect users who have retweeted or mentioned (or been retweeted by or been mentioned by) any adopter of the feature. Both these collections rely on Twitter’s `search/tweets` API endpoint, which is both free and rate-limited.⁴

With the adopter network constructed, for each feature, the tool then computes the following metrics over the network:

- (a) Cumulative distribution, over k , (abbreviated CDF_k) of percent of users who started tweeting about the feature after k or fewer network neighbors had done so.
- (b) Percent of all users who tweeted the feature before any of their network neighbors had done so, over time, or the Instigator Ratio.
- (c) Mean Tie Ratio (MTR)—Average density of connections among the first n adopters of the feature, for $n = 1$ to 100.
- (d) Number of adopters of the feature over time.
- (e) Average fraction of connections between adopter friends over time, termed the Mean Overlap Ratio.

3.5.2 Regional Contagion Monitor

For nominated hashtags, we calculated the same metrics, (a)–(e), described for the SCM. We base the CDF_k and MTR measures on the friend graph (as opposed to a dynamic retweets-mentions network in the case of the SCM) as it existed at the beginning of the given month, using the `friends/ids` query time stored with the edges. For each hashtag, the analysis period began at the start of the first day the hashtag had more than ten tweets, and ended on midnight of the trending day. We compute these measures for the current day.

In addition, we also looked at how users that adopted a hashtag were distributed across the network. We used the Louvain community detection algorithm [3]

⁴ <https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets>.

to define communities within the friend network. For each hashtag, we then computed

- (f) the *community entropy*, or the entropy of the counts of adopting users in each community.

Lower entropy values are associated with the adopting users being restricted to fewer communities; and higher entropy values are associated with users being spread out across more communities.

3.5.3 Productionalized Contagion Monitor

For the productionalized tool, we compute same five metrics as for the SCM, and do not compute communities.

3.6 Reporting

The Contagion Monitor generates a report of all metrics for each feature and also saves plots of them as PDFs. Some example plots are show in Fig. 1. See [7] for more detail on these metrics.

3.7 Coordination Framework

3.7.1 Introduction

Authors of [9] set out to formulate calculable, multi-dimensional signatures (metrics) of coordinated behavior in online social platforms to distinguish organic collective action from the coordinated (or inorganic). This framework relies on multiple metrics computed along three main dimensions: *network*, *temporal*, and *semantic*. Along each dimension, metrics target analysis at one of three levels: at the *event* level, at the *cluster* or community level, or at the *network* level.

These metrics capture ideas such as—coordinated activity has different network properties and dynamics as compared to spontaneous, organic activity. Other ideas include—participant engagement with the campaign over time is distinctly different between the two cases. Along the third dimension, the semantic, coordinated language is more likely to be focused on a single topic compared to authentic messaging. The 2017 work [9] identifies 14 different metrics along combinations of levels of analysis and dimensions of the framework. The implementation further computes metadata around these to add context for evaluating the metrics.

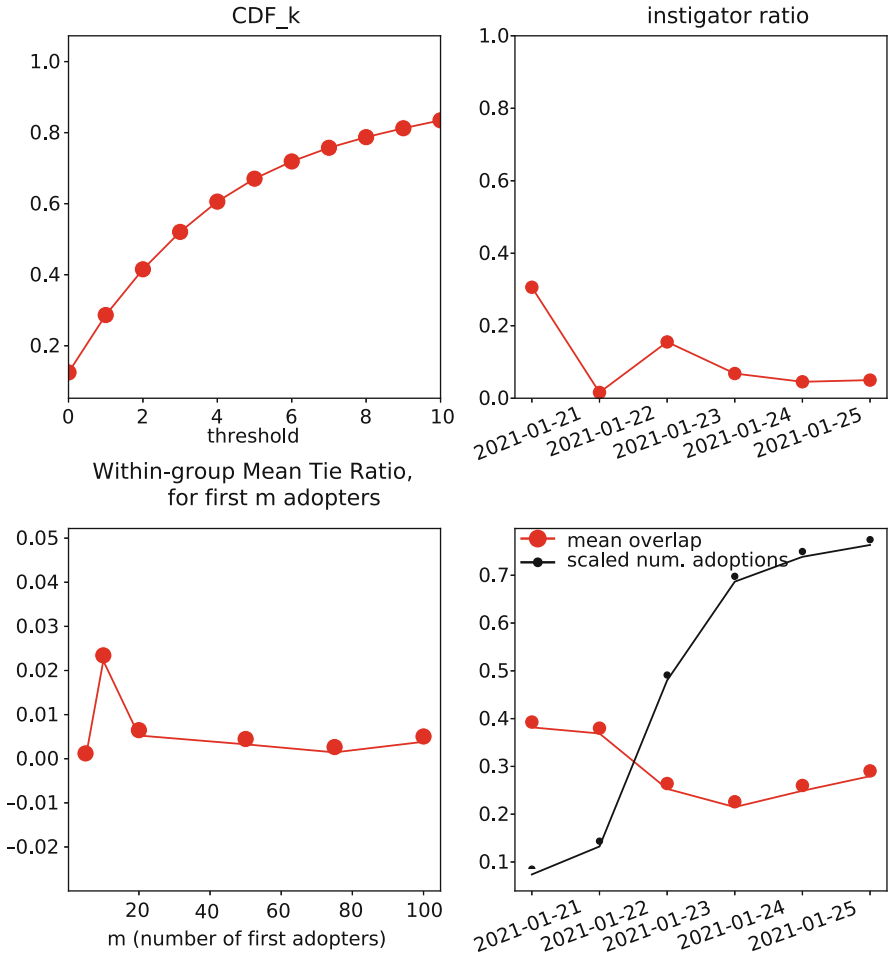


Fig. 1 Examples of Metrics (a)–(e) (see Sect. 3.5.1)

3.7.2 Implementation: Data Ingestion

These metrics are computed using content spreading over social networks—i.e., the content itself, the underlying network, and timestamps and measures of engagement with the content by the network nodes. The specific implementation leveraged in this integration with the Contagion MonitorTM is one that relies on pre-parsed data stores of [11], along with their network maps. As an illustration, if we want to know if hashtag #example is spreading in a coordinated way over a map named “Example Map”, then this framework will collect the usage data for the hashtag (every timestamp for each usage of the hashtag by a map member) along with the underlying network of connections between participants and the full text of their messages to compute all the metrics.

3.7.3 Implementation: Outputs

For each dimension, the framework generates dozens of pieces of metadata and metrics for the end-user to consider. The end-user must consider these manually, and spend the requisite time and attention to decide if they amount to coordination for the target hashtag. While there are several guidelines for the different metric and metadata combinations (e.g., fraction of activity originating from a single cluster being <0.3 is less likely to be a coordinated effort), as yet, these have not been folded up into heuristics that can offer automated judgements for the end-user. Such automated guidance is an improvement of this framework that we seek to complete in the future.

3.7.4 Integration

We have built in the capability for the Coordination Framework metrics to be computed for every candidate hashtag and URL selected by the Contagion Monitor. Currently, we are not pointing the Coordination Framework metrics calculation to content features other than hashtags or URLs (i.e., @mentions), as some infrastructure work for their implementation remains. Thus, for each candidate hashtag and URL that the Contagion Monitor finishes studying, we have two Contagion Monitor metrics, *complex contagion score* and *critical mass score*, and we also have the Coordination Framework metrics. Future work here will include implementation improvements such as allowing for greater level of control on which metrics are computed and the structure of the output (including the previously mentioned automated heuristic judgement outputs).

In the results section below, we show example outputs from recent runs of the Contagion Monitor with Coordination Framework metrics that illuminate the social spread of the candidate (hashtag) and the level of coordination behind it, and illustrate the usefulness of this combination.

4 Results

In this section, we mainly discuss our evaluations of the Contagion Monitor metrics in the different settings discussed thus far. In addition, we describe a nice result from the productionalized tool and also show some example cases of the combination of Contagion Monitor and Coordination Framework metrics before moving on to the Conclusions.

4.1 Framework for Evaluating the Contagion Monitors

We frame our evaluation under two key science questions, and an important question about data sizes for the productionalized tool. The CDF_k , MTR, and entropy metrics relate to first question, R1, and the remaining metrics relate to R2.

- R1. Do high-threshold social contagions spread through social networks in a fundamentally different way than low-threshold social contagions?
- R2. Do high-threshold social contagions “go viral” and begin to spread really quickly through social networks in certain conditions?
- R3. How little data can we collect and still achieve reliable and consistent judgements on complexity and virality?

For R1, the Contagion Monitors provide a formal encoding of how hashtags spread through the network, via metrics (a) and (c) in Sect. 3.5.1, but do not provide a human label for what a “high-threshold social contagion” is. For such human labels, we turn to previous research (Sect. 2.2), which suggests that political and social movements tend to spread like high-threshold contagions [4, 21]. This observation naturally suggests that we can evaluate the Contagion Monitors by generating human labels for whether a hashtag represents a political or social movement, and compare them against our threshold metrics (see Sect. 4.2.1 for more on this).

For R2, the Contagion Monitors provide a formal encoding of our theoretical assumptions about conditions when a contagion might “go viral,” via metrics (d) and (e) in Sect. 3.5.1. They do not provide a human label for what a “viral” contagion is. We have performed preliminary investigations of human labels for assessing contagion virality and found, not surprisingly, that humans do not successfully identify viral vs. non-viral contagions. As a specific test, we compared human labels of hashtags to the number of tweets using these hashtags and found no correlation. Therefore, we turn to an alternate approach for addressing R2 (not described before in [2]).

To address virality thoroughly can be a complicated task. In general, there is no formal characterization of what is meant when a contagion is said to be viral. There are however, reasonable expectations around virality that we can use to make some measurements. A basic assumption that we expect to hold, is that if we correctly identify viral contagions on Twitter, then they should receive notably higher engagement on the platform. In the particular context of our formulation, we say a viral contagion is one that has reached critical mass, and thus to be spreading widely throughout the network and eventually through its full extent. This strongly suggests that engagement on Twitter for those features we deem to have reached critical mass should notably exceed the engagement observed for features we deem to not have reached critical mass, depending on how close to the critical mass point (in time) we capture the engagement. This is the effect we seek to observe to validate our scoring of features for virality.

Question R3 is relevant to the operation of this tool in production, in two ways. In the first, finding a measure of the lowest amount of data that can work can help with speed and costs. Secondly, this question addresses a fundamental issue of sampling that applies when collecting data from Twitter using its `search/tweets`⁵ API endpoint; a limitation that applies to all tools built using this outlet. To address this question, we conduct a bootstrapped analysis of our contagion metrics over a range of sample sizes.

4.2 Evaluation of Complexity of Contagion (R1)

4.2.1 Mechanical Turk Annotation

We used Amazon Mechanical Turk (AMT) to obtain human annotations of the hashtags nominated by the SCM and the RCM. AMT allows researchers to post Human Intelligence Tasks (HITs), asking AMT workers (turkers) to perform a task for a small payment. It is a common tool for obtaining labeled data at scale, in our case, to assess labels for adoption threshold. For our HITs, turkers were asked to read a set of tweets corresponding to the date a hashtag was nominated, answer a number of questions about the topic (or topics) associated with the hashtag, and give their judgements about the use of the hashtag by Twitter users.

For the SCM, we restricted our analysis to hashtags that were associated with English language tweets since many turkers are English speakers and we wanted to avoid having to identify turkers fluent in other languages. Language identification was carried out by running a language identification tool⁶ on pseudo documents created by concatenating all of the tokenized tweets returned for a hashtag (Public Twitter API) after removing sentence punctuation, URLs, and emojis. We used language predictions for documents with 25 or more tokens and found 866 hashtags identified as English. For the RCM, we did no screening of language since most Twitter content we have observed from Nigeria has been in English or Nigerian Pidgin English.

For the SCM hashtags, we split the hashtags into equal-sized sets based on their likelihood of being a complex contagion. We created a “complex contagion score” (CCS) as follows:

$$CCS = \begin{cases} 0 & \log_{10}(\bar{\rho}_{\text{first 100 adopters}}) \leq -3 \cap CDF_k(k \geq 2) \geq 0.7 \\ 1 & \log_{10}(\bar{\rho}_{\text{first 100 adopters}}) > -3 \cup CDF_k(k \geq 2) < 0.7 \\ 2 & \log_{10}(\bar{\rho}_{\text{first 100 adopters}}) > -3 \cap CDF_k(k \geq 2) < 0.7 \end{cases} \quad (1)$$

⁵ <https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets>.

⁶ <https://github.com/saffsd/langid.py>.

The score is an integer that ranges from 0 (unlikely to be a complex contagion) to 2 (very likely to be a complex contagion). The set size was set to the number of hashtags that had a complex contagion score of 2. Hashtags with a score of two were the most interesting to us since they had a high observed adoption threshold and a spread within a dense initial network of users, both of these measures being theoretical indicators of complex contagion. There were 127 with a complex contagion score of 2. Since we were using three hashtags per HIT, the total number of hashtags of each set was 126 giving us 126 HITs for 378 hashtags. For the RCM, 648 were selected at random without regard to contagion scores.

We deployed the HITs and required nine assignments per HIT, meaning that we asked for judgments from nine different turkers. There is no convenient way to restrict how many HITs a turker does for a particular batch of HITs. This has consequences since a few prolific turkers could be responsible for a disproportionate number of annotations. Using qualifications (a means provided by AMT to control turker’s access to HITs) we implemented an approach using JavaScript and a custom RESTful interface that allowed us to disqualify a turker after they did a certain number of HITs. In this case, we restricted turkers to one SCM HIT and ten RCM hashtags.

Each HIT contained three hashtags, with each hashtag appearing on a separate page. At the top of each page there was a link to a Twitter Advanced Search query showing tweets containing the hashtag for the analysis period. Turkers were required to click on this link, which opened in a separate tab, and read a sample of the tweets using the hashtag before answering any of the questions. They were also required to answer all of the questions (shown in Table 1) before advancing to the next page or submitting the HIT. We paid \$1.80 per HIT (based on trial runs, a single HIT took about 10 min giving turkers an estimated hourly rate of \$10.80). Each HIT allowed for nine assignments to unique turkers.

We ran a total of 640 HITs and obtained annotations from 1084 turkers across both sets of hashtags. A pair of check questions determined whether turkers were paying sufficient attention to the task. Results from assignments where workers failed this check were not used. Overall, the average number of assignments used from each HIT (out of a possible nine assignments) was 8.20 (s.d. = 1.234) for the RCM hashtags and 7.89 (s.d. = 1.457) for the SCM ones. The average agreement (the maximum number of responses to a question—casting all questions as a binary choice—that were in agreement) was 6.83 (s.d. = 1.672) and 6.36 (s.d. = 1.692), respectively.

4.2.2 Analysis of Annotated Hashtags

From our sets of annotated hashtags we investigated the relationship between the contagion measures ($CDF_k(k \geq 2)$, MTR, and, for RCM, the community entropy) and the question responses obtained from AMT. We sought to learn how different ways of indicating threshold (from the responses) mapped onto the metrics. So, we specifically looked at regression models using the responses as dependent variables

Table 1 HIT questions

| | |
|-----|---|
| 1. | After reading the tweets, please provide a careful description of what the hashtag is about (text response) |
| 2. | What general topics does this hashtag reference? Check all that apply (Multiple choice ^a + optional text response) |
| 3. | Have you heard of this hashtag outside of this HIT? (Yes/No) |
| 4. | How often have you seen this hashtag used in a Tweet or Retweet? (not counting the tweets you read for this task) (five-point Likert scale: strongly disagree – strongly agree) |
| 5. | Does this hashtag get used in tweets from two or more groups with opposing messages? (Yes/No) |
| 6. | The hashtag is controversial (defined as “prolonged public disagreement or heated discussion”) (five-point Likert scale strongly disagree – strongly agree) |
| 7. | The hashtag expresses an opinion (five-point Likert scale strongly disagree – strongly agree) |
| 8. | The hashtag expresses an opinion that matches my own. (six-point Likert scale strongly disagree – strongly agree + no opinion expressed) |
| 9. | I find this hashtag funny (five-point Likert scale strongly disagree – strongly agree) |
| 10. | I find this hashtag interesting (five-point Likert scale strongly disagree – strongly agree) |
| 11. | I would be uncomfortable using this hashtag (five-point Likert scale strongly disagree – strongly agree) |
| 12. | I would be uncomfortable if my friends used this hashtag (five-point Likert scale strongly disagree – strongly agree) |
| 13. | I find this hashtag to be offensive (five-point Likert scale strongly disagree – strongly agree) |
| 14. | Using this hashtag would send a disturbing message (five-point Likert scale strongly disagree – strongly agree) |
| 15. | This hashtag is related to a political movement (five-point Likert scale strongly disagree – strongly agree) |
| 16. | Many Twitter users would be concerned about offending other users if they used this hashtag (five-point Likert scale strongly disagree – strongly agree) |
| 17. | Check all of the following reasons why someone might NOT want to use this hashtag (Multiple choice ^b + optional text response) |
| 18. | How contagious is this hashtag? (defined as spreading from one person to another) (four-point Likert scale ^c) |
| 19. | My friends would start using this hashtag... (choose the earliest option that is true) (six-point Likert scale ^d) |

^a See Appendix^b See Appendix^c See Appendix^d See Appendix

and the measures as independent variables to evaluate model fit and the significance of the measures. We then trained and evaluated classifiers for predicting positive question responses (as binary labels) using the measures as features.

4.2.3 Spam Filtering

While the SCM candidate hashtag selection infrastructure filters out spam-related hashtags, the RCM does not—other than initially filtering out hashtags spread exclusively by extremely productive or isolated, friendless accounts. We took the extra step of analyzing bot activity in the RCM data after noting the presence of hashtags with very low $CDF_k(k \geq 2)$ values (a high percentage of instigators) and low community entropy values (the accounts are isolated to a relatively small number of communities). Further examination of this data found that multiple accounts were tweeting the same links and hashtags simultaneously.⁷ A group of these accounts showed the same behavior over a number of hashtags and had a higher than expected connectivity within the friend network, suggesting they were part of a bot network. Forty-nine of the 648 hashtags were almost completely dominated by these accounts. They were removed, giving us a set of 599 hashtags for the following analysis, together with the hashtags from the SCM. The bot activity in the RCM dataset is of interest in and of itself, but will be left to follow-on research.

4.2.4 Linear Regression

We calculated the mean of the nine responses to each HIT question for both sets of hashtags. For the yes/no questions, we interpreted a mean value of 0.5 or more as a positive response; for the five-level Likert scale questions, a mean value of 2.5 or more was taken as positive. Questions with less than 5% positive responses were ignored.

Linear regression models were generated for each question using all three measures. For brevity, we focus on four responses: controversial and political (which should correspond to high-threshold contagions) and news events (labeled as “events”) and sports (which should correspond to low-threshold contagions). The results are in Tables 2 and 3.

The results show that $CDF_k(k \geq 2)$ has a positive and significant relationship with controversial and political hashtags, and a negative and significant relationship with sports and news events hashtags (except for the RCM monitor, where the measure has a positive and significant relationship with news events hashtags).

For both the monitors, the coefficients for MTR are negative and statistically significant. This result is in contrast with complex contagion theory, which indicates

⁷ Millisecond resolution is not available in the metadata returned by the search API, so ‘simultaneously,’ in this case, is at resolution of one second.

Table 2 Regression results for SCM hashtags

| Question | Intercept | 1-CDF($k = 2$) | Mean Tie Ratio | Adjusted R-squared |
|---------------|-----------|------------------|----------------|--------------------|
| Controversial | 1.77 | | | 0.00 |
| | 1.06 | 3.00*** | | 0.22 |
| | 1.12 | 3.38*** | -7.81*** | 0.25 |
| News events | 0.36 | | | 0.00 |
| | 0.37 | -0.05(0.55) | | 0.00 |
| | 0.38 | 0.03(0.74) | -1.54** | 0.03 |
| Politics | 0.42 | | | 0.00 |
| | 0.18 | 1.01*** | | 0.23 |
| | 0.21 | 1.18*** | -3.41*** | 0.29 |
| Sports | 0.09 | | | 0.00 |
| | 0.17 | -0.33*** | | 0.05 |
| | 0.18 | -0.31*** | -0.43(0.41) | 0.05 |

** p -value ≤ 0.01

*** p -value ≤ 0.005

Table 3 Regression results for RCM hashtags

| Question | Intercept | 1-CDF($k = 2$) | Mean Tie Ratio | Entropy | Adjusted R-squared |
|---------------|-----------|------------------|----------------|-------------|--------------------|
| Controversial | 1.05 | | | | 0.00 |
| | 0.91 | 0.55*** | | | 0.03 |
| | 0.91 | 0.94*** | -1.62*** | | 0.08 |
| | 1.45 | 0.83*** | -2.15*** | -0.39** | 0.10 |
| News events | 0.24 | | | | 0.00 |
| | 0.19 | 0.18*** | | | 0.03 |
| | 0.19 | 0.29*** | -0.47*** | | 0.08 |
| | 0.11 | 0.31*** | -0.39*** | 0.06(0.13) | 0.10 |
| Politics | 0.12 | | | | 0.00 |
| | 0.02 | 0.35*** | | | 0.03 |
| | 0.03 | 0.44*** | -0.37*** | | 0.08 |
| | 0.12 | 0.42*** | -0.46*** | -0.07(0.06) | 0.10 |
| Sports | 0.32 | | | | 0.00 |
| | 0.49 | -0.61*** | | | 0.03 |
| | 0.49 | -0.41*** | -0.84*** | | 0.08 |
| | 0.32 | -0.38*** | -0.68*** | 0.12(0.08) | 0.10 |

** p -value ≤ 0.01

*** p -value ≤ 0.005

that political and controversial hashtags would be more likely to emerge in denser network neighborhoods. We investigated this pattern further by visualizing the relationship between MTR and the hashtag labels. It follows the general pattern shown in Fig. 2.

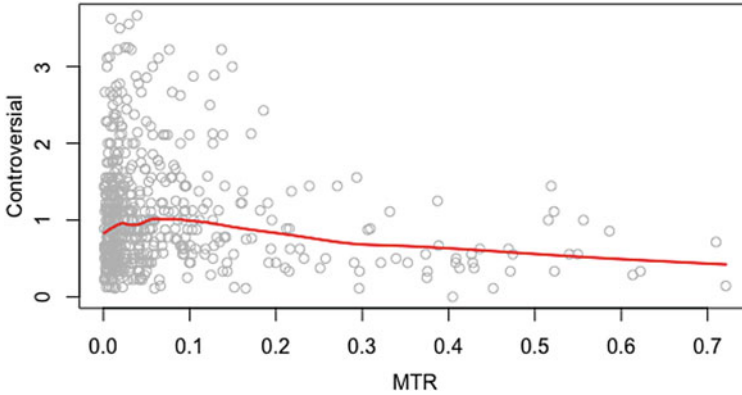


Fig. 2 Mean Tie Ratio vs. Controversial score for RCM

Table 4 Sample hashtags with high MTR

Streetmediapromotions, formularbyweflo, formulaoutsoon, doit, goosebumpsvideobyto, watchformulavideo, akwaibomtotheworld, formulavideo, lilayunchangeable, nozippy, pmfa2017, viktohybnl, smirnoffnightuyo, kissmebyoludre, sirehabbiibbpr, nammkpohmfo, theycypher, ipoetry, factswithkulqee, 2daystorepurclubuyo, afrima2017, smoothsummersplash, marryjuanabykamartachio, hypaft9ice

The pattern shows an increase in Controversial Score for $0 \leq MTR \leq 0.1$ and a rapid decrease in Controversial score thereafter. We manually examined hashtags identified by the RCM with $MTR > 0.1$. A sample is shown in Table 4. Many of these hashtags seem related to marketing or spam. While Fig. 2 does not suggest an explicit cutoff for MTR, these observations suggest that the cause of the negative coefficient for MTR in Tables 2 and 3 is a set of spam-related hashtags with extremely high mean tie ratios.

Finally, the results show that Entropy does not have a statistically significant relationship with any of the labels, except for controversial, where the relationship is negative.

4.2.5 Classification

We constructed a classifier for HIT questions, using $CDF_k (k \geq 2)$, MTR, and Community Entropy (where applicable). For the classification experiments we used a 70%–30% training–test split and trained a model using a support vector machine with a linear kernel. The positive classes (turkers said “yes” or agreed with the question) were the minority class for all questions. The average positive response rate was 0.28 (0.137) for the SCM hashtags and 0.16 (0.114) for the RCM. To address this class imbalance we used minority class oversampling using the SMOTE

algorithm [5]. We then ran the model against the held-out test set and calculated sensitivity (the true positive rate or recall), the specificity (the true negative rate), the accuracy, and the accuracy of a baseline classifier that always selects the majority label. This procedure was repeated on 10 random training–test splits and the performance measures were averaged across trials. For each trial we used a grid search to find the best value of C , the regularization term for the linear kernel. The results for a selection of the questions from the two sets are shown in Table 5. The table shows that for the SCM, $1 - CDF(k = 2)$ produces the best results by accuracy, though for the RCM, the inclusion of MTR as a feature provides an accuracy boost. Overall, both classifiers show best performance at predicting controversial and politics-related hashtag labels.

4.3 Analysis of Critical Mass for Virality (R2)

As previously mentioned, we try to identify whether a contagion has reached critical mass to see if it has gone viral. We assess critical mass using metrics (d), the adoption curve, and (e), the mean overlap ratio (Sect. 3.5.1). The characteristic signature that indicates the contagion has spread out of its local clusters and out to new communities (or having reached critical mass) is depicted in the X -shape visible in the bottom-right panel of Fig. 1, which shows these two metrics over-plotted in the same plot. As the contagion reaches newer communities, its network adoptions are on the rise. As the contagion spreads to new individuals outside of local clusters, there are fewer friendship ties between the new adopters and the previous ones. Thus, the average density of adopter friends (mean overlap ratio) decreases at break-out. This decrease in mean overlap ratio and the corresponding rise in adoptions produces the unique characterizing X -shape for a contagion spreading through critical mass.

To provide a useful heuristic of this signature, we capture the metrics into what we devise and call the *critical mass score*. The *critical mass score* (abbreviated as *CMS*) is formulated as shown below (let η_N be the total number of adoptions, and η_i be the initial number of adoptions, and also let \overline{ovp} represent mean overlap):

$$CMS = \begin{cases} 0 & \max\{\overline{ovp}\} \leq 1.5 \times \min\{\overline{ovp}\} \cup \eta_N \leq 3 \times \eta_i \\ 1 & \max\{\overline{ovp}\} > 1.5 \times \min\{\overline{ovp}\} \cap \eta_N > 3 \times \eta_i \\ 2 & \max\{\overline{ovp}\} > 1.5 \times \min\{\overline{ovp}\} \cup \eta_N > 3 \times \eta_i \end{cases} \quad (2)$$

To verify that these heuristic values well capture critical mass behavior, we obtain estimates of engagement volume for candidate hashtags and compare to their *CMS* values. In 2017, we selected 50 hashtags from a range of political and lower threshold topics. These contagions had *CMS* values ranging from 0 to 2, with about 20 hashtags with a score of 0, 17 with a score of 1 and 13 with a score of 2.

Table 5 Classification results for SCM and RCM hashtags

| Question | Model | SCM | | | | | RCM | | | | |
|---------------|----------------------------------|-------------|-------------|----------|-------------------|--|-------------|-------------|----------|-------------------|--|
| | | Sensitivity | Specificity | Accuracy | Baseline accuracy | | Sensitivity | Specificity | Accuracy | Baseline accuracy | |
| Controversial | 1-CDF($k = 2$) | 0.63 | 0.78 | 0.72 | 0.63 | | 0.45 | 0.68 | 0.67 | 0.95 | |
| | 1-CDF($k = 2$) + MTR | 0.62 | 0.76 | 0.71 | 0.63 | | 0.73 | 0.68 | 0.68 | 0.95 | |
| | 1-CDF($k = 2$) + MTR + ENTROPY | | | | | | 0.68 | 0.70 | 0.70 | 0.95 | |
| | 1-CDF($k = 2$) | 0.74 | 0.37 | 0.48 | 0.71 | | 0.44 | 0.72 | 0.68 | 0.87 | |
| | 1-CDF($k = 2$) + MTR | 0.57 | 0.68 | 0.87 | 0.87 | | 0.57 | 0.68 | 0.68 | 0.87 | |
| Sports | 1-CDF($k = 2$) + MTR + ENTROPY | | | | | | 0.62 | 0.62 | 0.62 | 0.87 | |
| | 1-CDF($k = 2$) | 0.81 | 0.52 | 0.54 | 0.93 | | 0.45 | 0.46 | 0.57 | 0.68 | |
| | 1-CDF($k = 2$) + MTR | 0.82 | 0.51 | 0.53 | 0.93 | | 0.73 | 0.49 | 0.61 | 0.68 | |
| | 1-CDF($k = 2$) + MTR + ENTROPY | | | | | | 0.68 | 0.55 | 0.66 | 0.68 | |
| | 1-CDF($k = 2$) | 0.61 | 0.76 | 0.69 | 0.53 | | 0.65 | 0.75 | 0.74 | 0.90 | |
| Politics | 1-CDF($k = 2$) + MTR | 0.61 | 0.76 | 0.69 | 0.53 | | 0.66 | 0.74 | 0.73 | 0.90 | |
| | 1-CDF($k = 2$) + MTR + ENTROPY | | | | | | 0.68 | 0.68 | 0.74 | 0.90 | |

Change in Tweets over 5-Day Period as a Function of Critical Mass

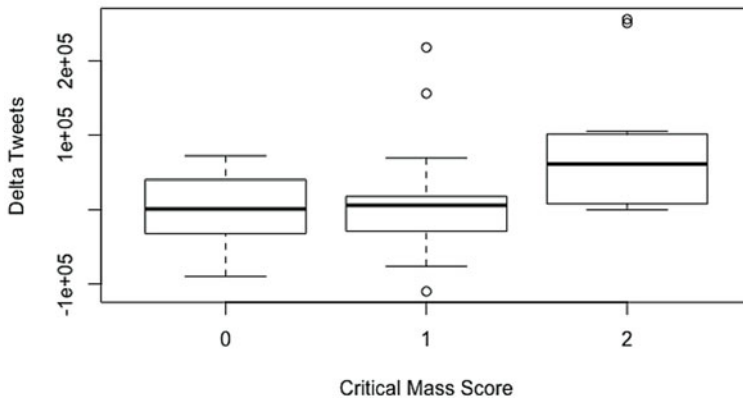


Fig. 3 All Twitter estimates of tweet volumes for contagions binned by their critical mass scores

To obtain engagement measures we turned to Twitter’s GNIP Historical Data service.⁸ GNIP allows clients to collect historical tweet data at a user specified level of sampling. Prior to submitting a collection request, GNIP offers an estimate of how many tweets a query is expected to return free of charge. This estimate, does not represent the accurate and exact number of tweets returned, but is instead meant to indicate the ballpark⁹ for the purpose of cost estimation. In side efforts, we have checked that the estimates provided by GNIP, particularly in the case of large collections, give a good estimate of the total number of tweets returned. We decided to use these estimates instead of collecting the data.

For each hashtag, we obtain a tweet volume estimate over the same five days over which we perform the critical mass score analysis. It is important to note here that the data used to compute critical mass, via the `search/tweets` endpoint, is necessarily a sampling of the full set of tweets engaging with the hashtag. There is low transparency on what fraction the full data is returned by this endpoint provides. The GNIP estimate is thus a better approximation of the size of the full engagement set (when requesting 100% sample in the query).

The resulting distributions of tweet-volume over the hashtags in each score-sample, are plotted in Fig. 3. The plot shows a distinct and significantly higher tweet volume for the tweets associated with contagions scoring a *CMS* value of 2. This clear result supports the formulation of *critical mass score* as a key component in assessing whether a social contagion has gone viral.

⁸ <https://developer.twitter.com/en/docs/twitter-api/enterprise/historical-powertrack-api/overview>.

⁹ <https://developer.twitter.com/en/docs/twitter-api/enterprise/historical-powertrack-api/overview>.

4.4 Analysis of Data Size (R3)

To address the question of how much data would be sufficient to get reliable estimates of both *CCS* and *CMS*, we perform a bootstrapped analysis. We selected a set of 20 features, @mentions, that we identified from our daily productionalized runs to be viral at some point in a two week period in early January of 2021. We used virality as a selection criterion to have a higher chance of finding viral contagions during our analysis period. Another benefit of selecting features on virality is that these get more engagement and thus more data volume, which reduces sampling instability (present at lower data volumes). These twenty features, being @mentions, were additionally selected for having high number of followers, again to reduce chances of low data sampling issues. The number of followers of the twenty features range from between tens of millions to about a hundred thousand.

For these 20 features, we collected the maximally available amount of data, for ten features at a time (as our run normally collects). We performed this collection in late January of 2021. Because this second data collection period is subsequent to the first, the distribution of complex contagion and critical mass scores in the following analysis is different than the distribution of scores when these features were first identified in early January 2021. We chose to run this test in our normal mode of operation with ten features at a time, instead of devising independent tests for each feature, given the added benefit this approach offers us in the way of evaluating our usual program. In general, the monitor will find different amounts of data for the different features depending on their relative engagement on Twitter.

We extract multiple bootstrapped sub-samples at different sampling fractions, drawn at the first level of data collection. This means, that we sample different fractions from the set of adopters identified, before the second round of retweets-mentions data collection, and then re-construct the network for the sample-adopters from the original data. We then compute our metrics and the *CMS* and *CCS* scores for each sample.

We examine the behavior of the metrics over the samples. The two scores over all features for the different samples are shown in Fig. 4. Overall, *CMS* does not seem to be very sensitive to sampling. We take a quick look at the ratio of total adoptions to initial adoptions; this is plotted in Fig. 5 for a subset of features. In general, this ratio is stable.¹⁰ The average overlap, the second component of *CMS*, also shows convergent behavior. We show this for a subset of features below in Fig. 6, and show the full set in the appendix. In particular, the overlap metric appears to stabilize somewhere between forty and sixty percent for the different features.

The complex contagion score seems to be more sensitive to sample fraction (Fig. 4), appearing to stabilize around forty percent. We take a look at the two main components for this score. Mean Tie Ratio, plotted in Fig. 7 for a couple of features, shows that some features show stabilization around 40 to 70%.¹¹ The more notable

¹⁰ See Appendix for all samples, including examples of features which show greater variation at lower sample fractions.

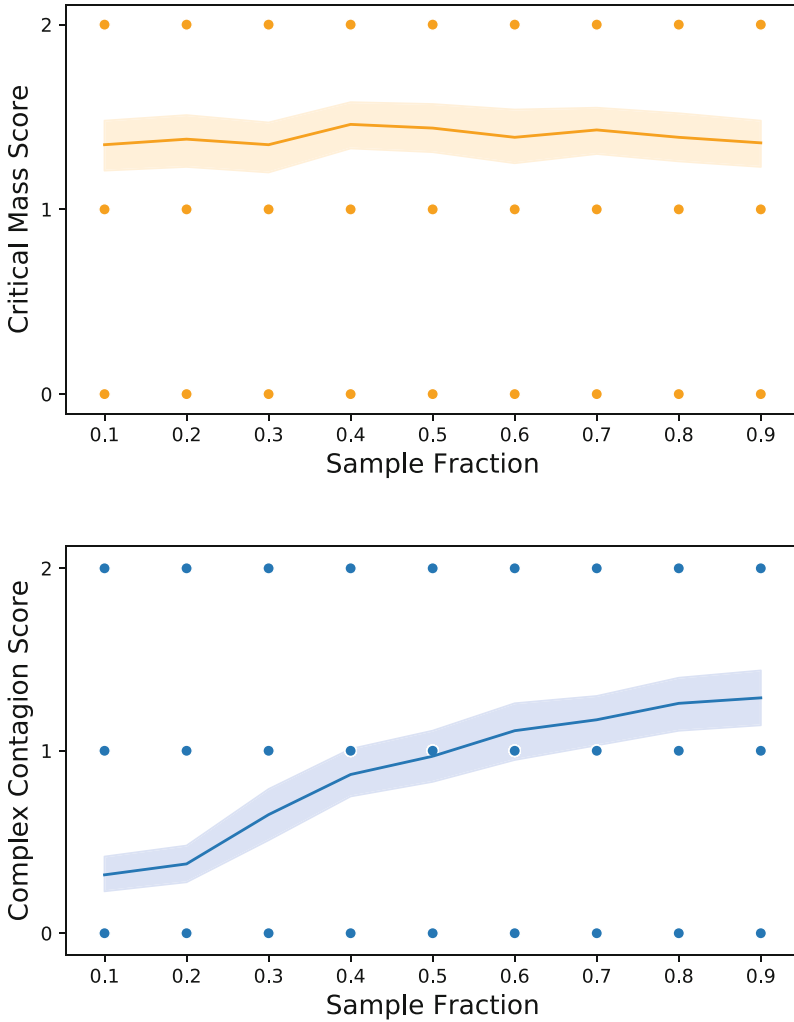


Fig. 4 Critical mass and complex contagion scores by sample, for all features. The shaded region represents the distribution of points along that direction

affect appears in the CDF_k metric, shown in Fig. 8. In particular Fig. 8 suggests that lower sample fractions are less likely to pick out complex contagions. This is not unsurprising, as it makes sense that randomly sampling from a network would result in a reduced signal at higher degrees. This figure suggests that one direction for further study is to use the ratio of CDF_k to the number of engaging nodes in

¹¹ See Appendix for plots for all features.

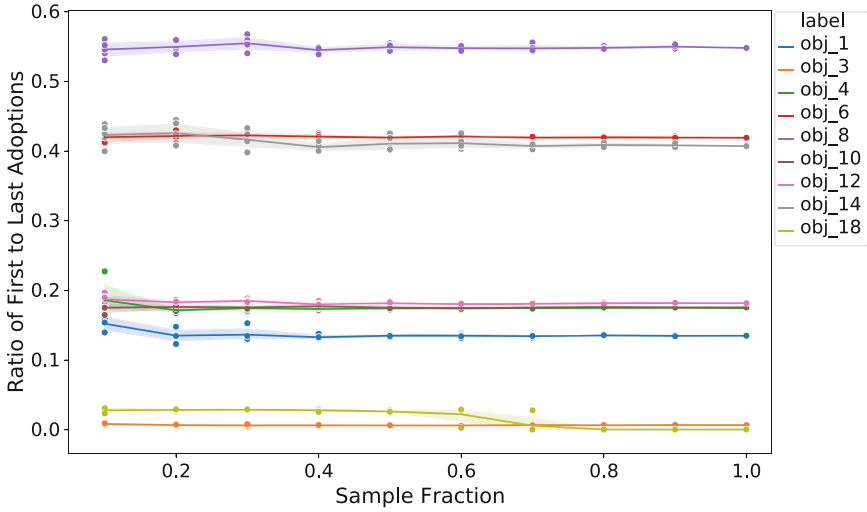


Fig. 5 Ratio of total adoptions to initial adoptions, shown for a random subset of features

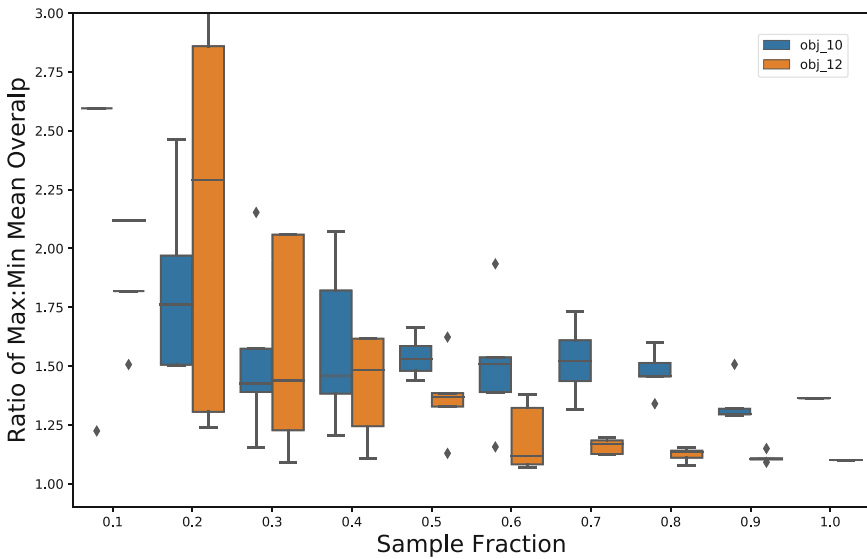


Fig. 6 Showing distribution of mean overlap values at each sample fraction for two features

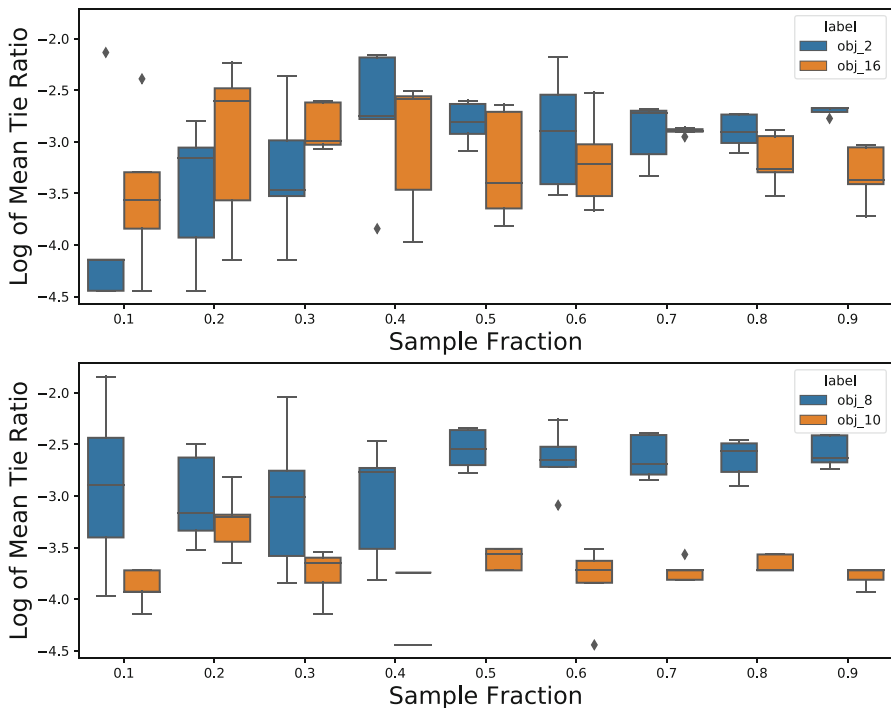


Fig. 7 Log of mean tie ratio of early adopters for a subset of features for different samples

the *CCS* score. We leave that to further investigation, along with the task of finding ways to mitigate this effect in the *CDF_k* metric.

4.5 A Fruitful Case Study

One of the nice benefits of using structural information to find emerging movements is that it offers the possibility of early detection (before count statistics catch up from adoptions). This is illustrated nicely by one particular discovery made by run of the productionalized Contagion Monitor™ of two rising artists going from fringe to mainstream. Graphika [11] was contacted by a client with a request to identify emerging artists at the South By Southwest (SXSW) Music festival in 2018, by studying the Twitter activity. To accommodate this request, [11] added the ability to study Twitter @mentions to the productionalized monitor. In March of 2018 we analyzed social contagions in a commercial environment—Twitter activity around the South By Southwest (SXSW) music festival, to look for up and coming musicians and artists (where supporting a new artist has a higher social cost due to their lower popularity). We found the Twitter handles of Desus and Mero, a

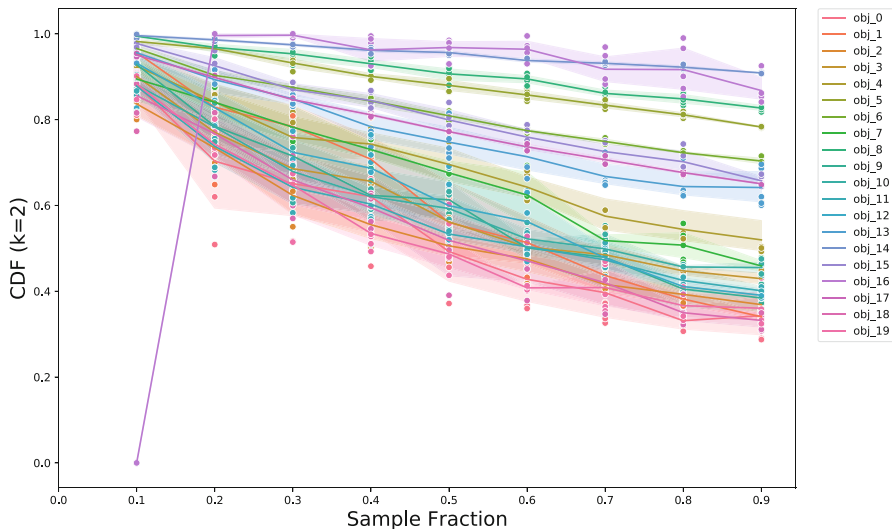


Fig. 8 CDF_k metric for all features over different samples. Zero values for CDF_k are artificially set when the distribution is incomplete

comedian and a DJ, had both a high adoption threshold and had reached critical mass. We followed these artists over a longer time period, and observed how their rapid spread on social media was followed by commercial success[12], including a successful performance tour and a dedicated program on the Showtime channel. This performance of the Contagion MonitorTM suggests that it can be a powerful tool for identifying both politically and commercially relevant content in social media.

4.6 Results with Coordination Framework Metrics

The above case study illustrates the power of the Contagion Monitor, but this Monitor is vulnerable to artificial or orchestrated attempts to spread information and dis-information, in that it will also surface those movements. And so, we add in the Coordination Framework metrics to assess the possibility of fabrication. The usefulness and validation of the Coordination Framework metrics themselves is discussed in [9]. Here, we focus on the integrated combination of these metrics along with the Contagion Monitor metrics.

For simplicity we only show a subset of the 14 metrics, from the *network* and *temporal* dimensions—as these are commonly used and easily illustrate the power of the combination. Two key metrics, that are often used as initial filters are the *Peakedness* and *Concentration*. These and the remaining presented are described next for understanding the results—more information about them is available in [9].

Peakedness is a temporal metric at the event level, and checks what fraction of activity occurred within 24 hours of the peak in activity. Organic conversations show asymptotically increasing and decreasing activity resulting in a cusp, and so shows a large fraction of the activity occurs within the peak. For inorganic activity, observed in [11] maps, the *Peakedness* metrics tends to be below 0.2, or that less than 20% of the activity happens during the peak.

Concentration is a network dimension metric at the community level of analysis. It measures the fraction of activity that arises out of one community. In the attention based clusters in [11]’s maps, coordinated activity is often more concentrated in a single cluster with about 30% or more of the activity arising from this cluster.

A first high level result, considering just the above two metrics, is that we see that less than 6% of the hashtags and URLs with sufficient counts, identified from [11]’s live map library over the last two years, present with *Peakedness* below 0.2 and *Concentration* above 0.3. This suggests a coordination has a low prevalence in the wild.

Metrics shown in Table 6 include *Concentration* and *Peakedness*, the median count per user (Med., in the table), Mean count (Mean) and standard deviation (Std.). In addition, we present two measures of the *Commitment* by an actor. Firstly, the average *Commitment to engagement* by an actor (M_E), or how many times on average a user uses the hashtag (or feature) before stopping altogether. Secondly, the average *Commitment in time* by an actor, which is the average length of time (last day–first day) in days that a node engages with the hashtag or URL.

We show two examples of coordinated contagions, and one example of an uncoordinated one. The first coordinated contagion is the hashtag “rest in peace Reza Shah” (in Persian), in support of Reza Shah of Iran, did not have sufficient activity to determine a critical mass and complex contagion score. Usage of the hashtag in the sample analyzed is concentrated in a cluster of Iranian monarchist sympathizers. *Peakedness* for the hashtag is 0, indicating an insignificant level of activity to establish a distinct peak.

The second coordinated contagion is the hashtag #b0105, in support of a protest in Germany on May 1st 2021, is spreading socially with reinforcement and is going viral according to its Contagion Monitor metrics. Usage of the hashtag in the sample analyzed is concentrated in a cluster of German Anti-fascist organizations and activists. *Peakedness* for the hashtag is 0, indicating an insignificant level of activity to establish a distinct peak.

The uncoordinated contagion example is #coronavirus, capturing the discussion about the COVID-19 virus. The hashtag is spreading socially with reinforcement and is going viral according to its Contagion Monitor metrics. Usage of the hashtag in the sample analyzed is concentrated in a cluster of Brazilian journalists. The activity around the hashtag is highly peaked, with over half of the activity in the sample occurring in one day.

The uncoordinated contagion also has a higher mean count of daily activity and a higher M_E than either of the coordinated contagions, but these differences could be due to the much higher, global, volume of conversation about COVID-19 as

Table 6 Coordination framework metrics to pick out coordinated, potentially inauthentic contagions

| Label | Name | Date | CCS | CMS | Med. | Mean | Std. | M_E | M_T | Pk. | Conc. |
|----------------|---------------------------|-------------|-----|-----|------|-------|-------|-------|-------|----------------|-------|
| Coordinated | “rest in peace Reza Shah” | Feb-09-2021 | n/a | n/a | 56 | 28.50 | 39.24 | 2.60 | 0.03 | 0 ^a | 0.49 |
| Coordinated | b0105 | May-04-2021 | 2 | 2 | 10 | 10.80 | 10.44 | 3.11 | 0.43 | 0 ^a | 0.43 |
| Un-coordinated | Coronavirus | May-16-2021 | 2 | 2 | 44 | 70.00 | 72.82 | 5.44 | 0.34 | 0.57 | 0.05 |

^a 0-value is used to indicate an insignificant amount of activity
n/a—indicates data not available (see text for details)

a topic relative to the more regional discussions around either of the coordinated contagions.

We would like to emphasize that all three examples, as any examples surfaced from the Coordination metrics section of the Contagion Monitor, require human analyst review for additional verification and context. However, the metrics provided by the monitor can greatly accelerate this analytic process by automatically suggesting candidates for further research.

5 Discussion and Conclusion

The regional (RCM) and streaming (SCM) implementations of the social Contagion Monitor that we have built successfully process a large volume of data to identify emerging hashtags, and other features, that are representative of low- and high-threshold contagions. These tools let us test complex social theories at a large scale and over long periods of time. We hope to extend the monitors further to collect other public data sources and integrate additional social theories, such as homophily, into their analytic toolkit.

Our evaluations of the RCM and SCM done using hashtag test data and a label classification, confirm that hashtags with more social reinforcement are more likely to be labeled as political and controversial (Sect. 4.2). The streaming Contagion Monitor also shows that hashtags with less social reinforcement are more likely to be labeled as news events and sports. These findings are in precise accordance with complex contagion theory [4]. The regional Contagion Monitor diverges from this pattern mildly to show that hashtags with more social reinforcement are also more likely to be labeled as news events. It is interesting to consider whether sharing news events is a higher-threshold behavior in Nigeria than globally, or whether a different confound accounts for this pattern.

We have observed a surprising negative relationship between Mean Tie Ratio and controversial and political hashtags (Sect. 4.2). This relationship is inconsistent with complex contagion theory, which states that complex contagions are more likely to arise in more dense neighborhoods. However, a closer investigation of the relationship shows that it is driven by hashtags with extremely high MTR, which may be spam related. We do not see this finding as ultimately at odds with complex contagion theory, but an interesting development of the same. Perhaps organizers of spam related hashtags form dense networks to facilitate the spread of content due to social reinforcement.

Our classification results (Sect. 4.2) demonstrate that (a) both monitors perform best when predicting labels of controversial and political hashtags, and (b) most accuracy gain comes from including $1 - CDF(k = 2)$ as a feature, with MTR and Entropy contributing far less to accuracy gain. These results are also consistent with our interpretation of complex contagion theory: we use CDF_k as a proxy measure for contagion threshold, so it is the key differentiator between complex and simple contagions. The other two metrics provide circumstantial evidence for contagion

threshold, as complex contagions can leverage redundant ties more easily in adopter networks with higher MTR and lower community entropy. Since we only evaluated linear classifiers, it is possible that higher-dimensional classifiers may achieve more accuracy gain from these measures. We leave this investigation for future work.

The classification performance of the streaming Contagion Monitor beats the baseline for controversial and political hashtags using only a linear classifier and two or three features. To our knowledge, our findings have never been replicated with the same instrument across multiple social, cultural, and linguistic settings and our paper is also the first to label hashtag categories and evaluate complex contagion theory in the context of a productionalized tool. Overall, we have found the social Contagion Monitors to successfully identify emerging low- and high-threshold movements in both regional and global Twitter settings.

We were able, in this extended work, to show using estimates of tweet volume collected through the GNIP service, that our formulation of the *critical mass score* at a value of 2 indeed picks out contagions that receive statistically higher engagement on platform compared to features that receive lower *critical mass scores* (Sect. 4.3). We consider this a first step toward confirming critical mass as a virality indicator in the empirical SCM (also productionalized) setting, and recognize the opportunity for additional methods for evaluating and validating this method.

We further investigated the stability of the two scores over samplers samples of the collected data for twenty features (Sect. 4.4). We learned that *CMS* is generally stable over the data sizes and sample fractions we used in this study. We additionally observed that *CCS* exhibits a linear relationship, with a lesser probability of detecting complex contagion at lower sample fractions. We must leave to future efforts, the task of identifying how and why different data sets exhibit different slopes for this pattern, and the additional related task of mitigating the slight affect that this introduces in the *CCS*. Our current approach thus shows successful applications of both *complex contagion score* and *critical mass score*, with room for more refinement in the *complex contagion score*.

We additionally folded in the capability to help weed out potentially inorganic events and movements that appear to spread in a viral manner from person to person. We find that in the pre-existing real world network and content collections of [11], a low fraction, 6%, present as coordinated and engineered contagions, according to the Coordination Framework [9] metrics. These metrics offer the opportunity to manually consider multiple relevant pieces of information in assessing whether an event is organic or not, and no one single metric would be able to provide this judgement with accuracy across all scenarios. This process has room for improvement, in at least the building of heuristics to help reduce the burden on manual examination.

One shortcoming of our approach, beyond the inherent limitations of using “digital traces” to model diffusion in social networks, is that we model complex contagions as separate events. Movements that lead to behavioral change, however, can inspire one another, requiring them to be modeled together and not in isolation. We leave such modeling efforts for future study. We recognize again that this tool may surface dis- or mis-information instead of real transformative movements,

resulting from our focus on data quantity over data quality. We have mitigated this risk some by merging in metrics from the Coordination Framework [9] to identify coordinated information spread. Reliable classification of dis-/mis-information remains a problem at large, and would be a useful addition.

We hope in future work, to combine these with methods for verifying the quality of an information stream, either in automation [9] or along with expert human review. We additionally hope to expand the capability with new data sources and analytics. All in all, the performance of the Contagion Monitor measured here and in prior work suggests that it can be a powerful tool for identifying both politically and commercially relevant viral content in social media.

Appendix

See Tables 7 and 8 for topics and Likert scale descriptions for questions listed in Table 1.

See Figs. 9, 10, and 11 below for metrics on all features, as relate to the analysis in Sect. 4.4. Additionally, Fig. 12 shows variance at each sample fraction for the different metrics (over the bootstrapped samples). Finally, in Table 9 we report metadata for the @mentions used in the data volume study.

Table 7 Question topics from Table 1

| Question | Topics |
|----------|---|
| Q-2 | Movie; TV; News Event; Music; Celebrity; Sport; Health; Religion; Marketing Campaign; Hacking or Cyberattack; Politic; Divisive Social or Moral Issue; Sexuality; Corruption; Injustice; Crime/Police; War or Armed Conflict; Protest or Rally; Criticism of Society; Criticism of Government; Meme; Joke |
| Q-17 | Offensive; Disturbing; Polarizing; Using this hashtag could hurt one's reputation; Unfamiliar; Uninteresting; The hashtag uses inappropriate language; The hashtag is used by people that one might not want to be associated with |

Table 8 Likert scales for questions in Table 1

| Question | Scale descriptions |
|----------|---|
| Q-18 | 1. Not contagious at all—People would not use this hashtag no matter how many others were using it |
| | 2. Slightly contagious—People would only use this hashtag if they saw many people using it |
| | 3. Moderately contagious—People might decide to use this hashtag if they noticed a few people were using it |
| | 4. Highly contagious—People would use this hashtag as soon as they saw someone use it for the first time |
| Q-19 | 1. If they heard about it from any random source |
| | 2. If one of their friends was using it |
| | 3. If a few of their friends were using it |
| | 4. If many of their friends all were using it |
| | 5. If nearly all of their friends were using it |
| | 6. Under no circumstances |

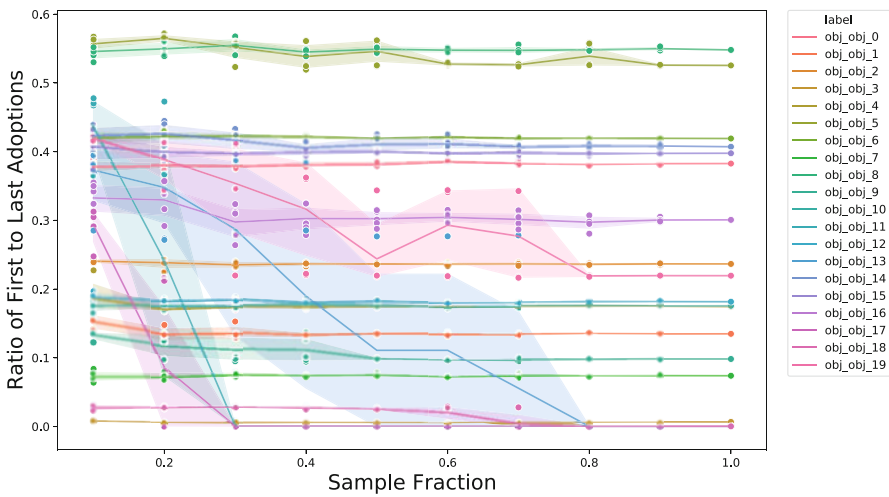


Fig. 9 Ratio of total to initial adoptions for all features over sample fraction

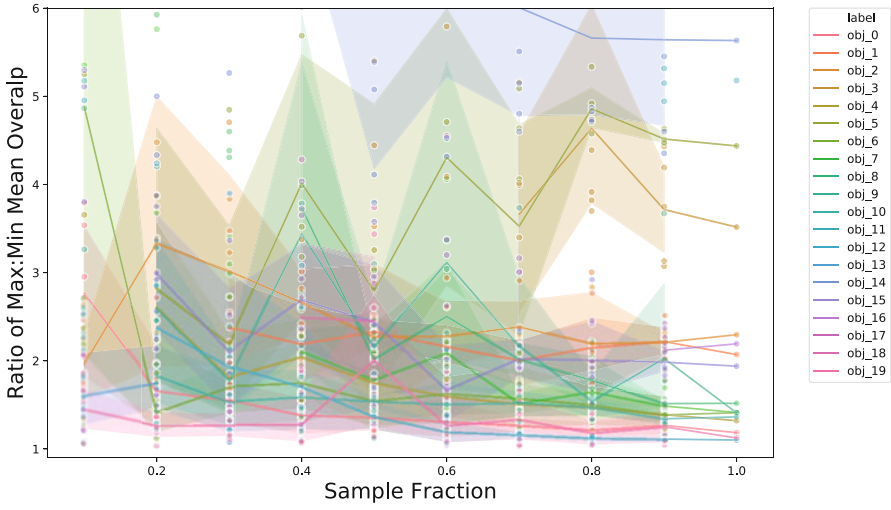


Fig. 10 Ratio of max to min overlap over sample fraction for all features

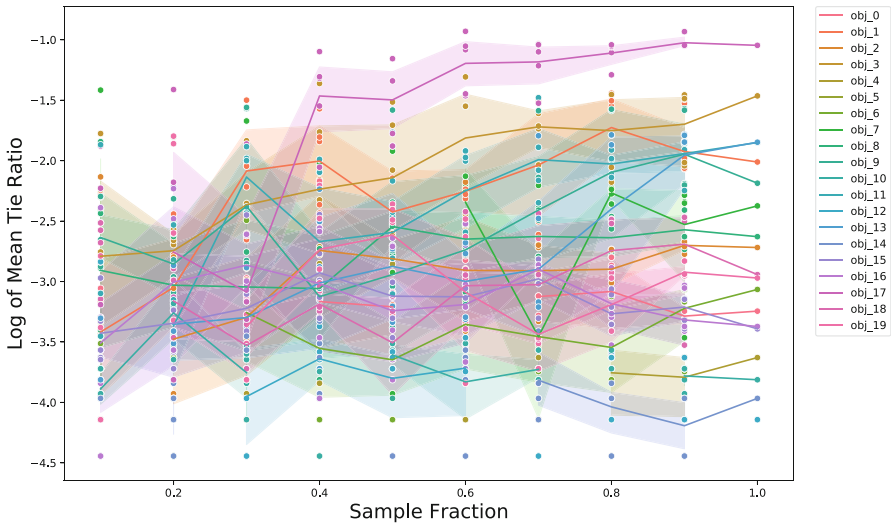


Fig. 11 Log of mean tie ratio of early adopters over sample fraction for all features

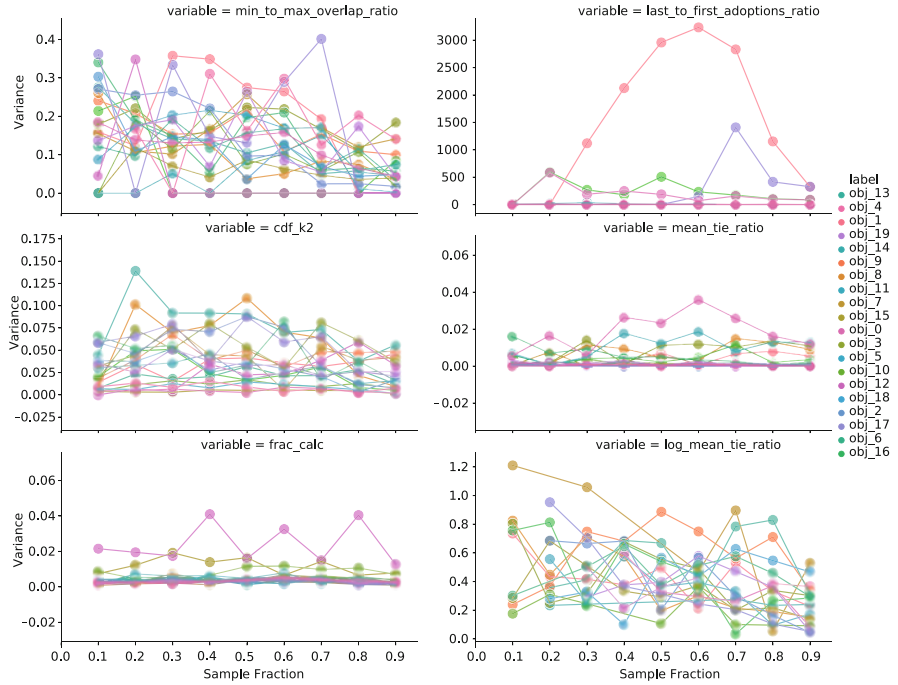


Fig. 12 Variance of bootstrapped samples at each sample fraction for multiple metrics and all features. “frac_calc” is the calculated sample fraction value (from network size of re-sampled and original data). In general, variance of many variables either decreases or remains steady with sample fraction, except for a few features. Some of the more variable features also interestingly show a varying calculation of sample-fraction

Table 9 @mentions metadata at max data

| Label | num_tweets | num_users | Followers |
|--------|------------|-----------|------------|
| obj_0 | 31,067 | 21,391 | 2,800,000 |
| obj_1 | 33,048 | 20,154 | 82,000 |
| obj_10 | 36,166 | 24,451 | 887,000 |
| obj_11 | 91,114 | 51,102 | 290,000 |
| obj_12 | 38,988 | 26,412 | 1,300,000 |
| obj_13 | 84,795 | 58,239 | 16,400,000 |
| obj_14 | 11,424 | 7133 | 4,200,000 |
| obj_15 | 27,861 | 21,844 | 115,000 |
| obj_16 | 1741 | 1024 | 150,000 |
| obj_17 | 81,988 | 58,071 | 50,000,000 |
| obj_18 | 96,893 | 40,011 | 1,400,000 |
| obj_19 | 98,837 | 59,448 | 665,000 |
| obj_2 | 61,194 | 36,891 | 1,300,000 |
| obj_3 | 30,944 | 20,049 | 10,500,000 |
| obj_4 | 66,957 | 46,867 | 115,000 |
| obj_5 | 45,956 | 30,654 | 280,000 |
| obj_6 | 72,068 | 48,355 | 18,500,000 |
| obj_7 | 17,195 | 14,233 | 816,000 |
| obj_8 | 7679 | 6241 | 958,000 |
| obj_9 | 19,025 | 13,652 | 410,000 |

References

1. Barash V, Cameron C, Macy M (2012) Critical phenomena in complex contagions. *Soc Netw* 34(4):451–461. <https://doi.org/10.1016/j.socnet.2012.02.003>, <http://www.sciencedirect.com/science/article/pii/S0378873312000111>
2. Barash V, Fink C, Cameron C, Schmidt A, Dong W, Macy M, Kelly J, Deshpande A (2020) A twitter social contagion monitor. In: Accepted to IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM) proceedings
3. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *J Stat Mech Theory Exp* 2008(10):P10008. <http://stacks.iop.org/1742-5468/2008/i=10/a=P10008>
4. Centola D, Macy M (2007) Complex contagions and the weakness of long ties1. *Am J Sociol* 113(3):702–734
5. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
6. Cheng J, Adamic L, Dow PA, Kleinber J, Leskovec J (2014) Can cascades be predicted. In: Proceedings of 23rd international world wide web conference
7. Fink C, Schmidt A, Barash V, Cameron C, Macy M (2016) Complex contagions and the diffusion of popular twitter hashtags in Nigeria. *Soc Netw Anal Min* 6(1)
8. Fink C, Schmidt A, Barash V, Kelly J, Cameron C, Macy M (2016) Investigating the observability of complex contagion in empirical social networks. In: Proceedings of the 10th international conference on weblogs and social media
9. François C, Barash V, Kelly J (2017) Measuring coordinated vs. spontaneous activity in online social movements. Preprint available on SocArxiv. <https://osf.io/aj9yz/>

10. González-Bailón S, Borge-Holthoefer J, Rivero A, Moreno Y (2011) The dynamics of protest recruitment through an online network. *Sci Rep* 1(197)
11. Graphika (2017) Graphika. <http://www.graphika.com>
12. Graphika (2019). <https://graphika.com/posts/the-desus-and-mero-story/>, <https://graphika.com/posts/the-desus-and-mero-story/>
13. Health Services H (2017) Office for human research protections. subpart d. Additional protections for children involved as subjects in research. <https://www.hhs.gov/ohrp/regulations-and-policy/regulations/45-cfr-46/index.html#subpartd>
14. Human Rights Watch (2016) “tell me where i can be safe” the impact of Nigeria’s same sex marriage (prohibition) act. <https://www.hrw.org/report/2016/10/20/tell-me-where-i-can-be-safe/impact-nigerias-same-sex-marriage-prohibition-act>
15. Hexagon C (2017) Crimson hexagon. <http://www.crimsonhexagon.com>
16. Indiana University(2017) Observatory on social media. <http://truthy.indiana.edu/>
17. Kelly J, Barash V, Alexanyan K, Etling B, Faris R, Gasser U, Palfrey J (2012) Mapping Russian twitter. *Berkman Cent Res Publ* (3)
18. Madden M, Lenhart A, Cortesi S, Gasser U, Duggan M, Smith A, Beaton M (2013) Teens, social media, and privacy. Pew research center internet, science and tech. <http://www.pewinternet.org/2013/05/21/teens-social-media-and-privacy/#fn-67-1>
19. Morris S (2000) Contagion. *Rev Econ Stud* 67:57–78
20. Nahon K, Hemsley J. (2013) Going viral. *Polity*
21. Romero DM, Meeder B, Kleinberg J (2011) Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: *Proceedings of the 20th international conference on world wide web*. ACM, New York, pp 695–704
22. Salesforce (2017) Social studio by salesforce. <https://www.salesforce.com/products/marketing-cloud/channels/social-media-marketing/>
23. Stanford University (2017) Nifty. <http://snap.stanford.edu/nifty/>
24. Traag VA (2016) Complex contagion of campaign donations. *PLoS One* 11(4)
25. Watts DJ, Strogatz SH (1998) Collective dynamics of ‘small-world’ networks. *Nature* 393:440–442

TenFor: Tool to Mine Interesting Events from Security Forums Leveraging Tensor Decomposition



Risul Islam, Md Omar Faruk Rokon, Evangelos E. Papalexakis,
and Michalis Faloutsos

Abstract How can we have a security forum to “tell” us its activities and events of interest? We take a unique perspective which is to identify these activities without any a priori knowledge. This is a key difference compared to most of the previous problem formulations. While most of the previous efforts are usually searching for specific information, mining security forums to extract useful information has received relatively little attention despite of some recent efforts. We propose TenFor, an unsupervised tensor-based approach, to systematically identify important events in a three-dimensional space. The dimensions are (a) user, (b) thread, and (c) time. Our method consists of the following three high-level steps: (a) a tensor-based clustering across the above-mentioned three dimensions, (b) an extensive cluster profiling using both content and behavioral features, and (c) a deeper investigation to identify key users and threads within the events of interest. Additionally, we implement our approach as a powerful and easy-to-use platform for practitioners. In our evaluation, we find that 83% of our identified clusters capture important events and we find more important clusters compared to the previous approaches. Our approach and platform constitute an important step towards detecting activities of interest from a forum in an unsupervised learning fashion in practice.

Keywords Tensor decomposition · Security forums · Event extraction

1 Introduction

Online security forums have emerged as a platform where users generally initiate a discussion about their security-related issues. Therefore, security forums contain a wealth of information that currently remains unexplored. These forums aggregate valuable information in an unstructured way and initial work argues for a wealth of

R. Islam · Md O. F. Rokon · E. E. Papalexakis · M. Faloutsos (✉)

UC Riverside, Riverside, CA, USA

e-mail: risla002@ucr.edu; mroko001@ucr.edu; epapalex@cs.ucr.edu; michalis@cs.ucr.edu

useful information, for example, emerging threats and attacks, promotion of hacking skills, and technical tutorials. Discussions around these topics at one or more points in time often involve a large number of users and threads, and we can think of them as important **events** in the life of the forum.

Problem Definition How can we spot major events of interest in a forum in an unsupervised fashion? The input is a forum and the desired outputs are the major events that capture the key activities in the forum and could be of interest to a security analyst. For example, a security analyst would want to identify outbreaks of attacks, the emergence of new technologies, groups of hackers with tight and focused interests, and underground black markets of hacking services. The challenges are that the information is unstructured and that we want to do this in an unsupervised way. Basically, **we want the forum to “tell” us its events of interest.**

Related Works Although security forums hide a wealth of information, mining these forums has received relatively little attention and only recently. We can identify the following three main categories of related efforts: (a) security forum related studies, (b) analysis of blogs, social media, and other types of forums, and (c) tensor-based mining approaches. We discuss these efforts in Sect. 6.

Contribution We propose and develop TenFor, a systematic tensor-based approach and tool, to identify important events in an unsupervised way in a forum as our key contribution. Our approach operates at the three-dimensional space of (a) user, (b) thread, and (c) time. Our method consists of the following three main steps: (a) *clustering* using a tensor decomposition, (b) *profiling* using both content and behavioral metrics, and (c) *investigating* using an automated, but customizable, method to capture the dynamics of the clusters and provide an interpretable view.

Novelty We summarize the novelty of our approach in terms of techniques and features as follows: (a) it operates in an unsupervised way, (b) it adapts and combines tensor-based clustering, behavioral profiling, and NLP methods, (c) it is user-friendly by being parameter-free with *optional* tuning of the parameters by the end-users so that they can adjust the granularity and information detail of the results, and (d) it provides visual and intuitive fingerprints of the events of interest. All these capabilities are further discussed later.

Overall, an end-user can obtain the following results: (a) the most dominant clusters in the lifespan of the forum, (b) profiling information about these clusters including key users, key threads, key dates, and key topics and keywords, (c) optional labeling of the clusters using user-specified keywords. Visually, the results can appear in a StoryLine View or a Table View as shown in Fig. 1 and Table 3 respectively.

In our evaluation, we apply TenFor on three security forums, one gaming forum, and GitHub malware dataset with a total of 58254 users. We find that **83%** of our identified clusters are meaningful. For example, they exhibit high intra-cluster thread similarity, and the clusters revolve around interesting events as validated by



Fig. 1 StoryLine View: a user-friendly visualization of a cluster summary with our TenFor tool. We present one of the identified clusters, which captures the emergence of SimpleLocker ransomware from the Offensive Community forum

experts, crowdsourcing, and other methods. Our approach also compares favorably with previous approaches [2, 32] leveraging the power of tensor decomposition to strike the balance between size and number of clusters.

Going Beyond the Security Forums TenFor can be used on other types of forums although we focus primarily on security forums here. As a proof of this, we apply our approach on online gaming forum and GitHub malware dataset. We find interesting activities from there as well, including revenge hacking and romance scamming, surging of ransomware malware development as we discuss later.

The Overarching Vision We develop a powerful user-friendly platform that will be useful to both researchers, and industry practitioners as a tangible contribution. Our ambition is to make this platform a reference tool for forum analysis and inspire subsequent research and development.¹ The proposed hands-free event extraction is a significant capability where we let the forum to tell us what are the key activities of interest, namely “taking the pulse” of the forum. This can enable practitioners to shift through a large number of forums of interest efficiently and effectively. In the future, we will extend our tool by providing additional user-centric and content-centric capabilities.

Lineage This paper is an extended version of our earlier 8-pages paper [14]. We outline the key additions and changes in this version. First, we describe our method more thoroughly and clarify the reason behind different choices in our clustering algorithm, for example, Fig. 5. Second, we add one more dataset, GitHub malware

¹ Code repository: <https://github.com/RisulIslam/TenFor>.

dataset, in our “Results and Evaluation” section and we provide more detailed results for our other datasets, for example, Fig. 6. Third, we discuss the sensitivity of TenFor to the model parameters summarized in Table 5 and Fig. 9. We also provide step-by-step directions for the end-users to tune the model parameters in “Discussion” section. Fourth, we add another layer of detailed evaluation using synthetic data in Sect. 4.3. We compare TenFor with the current state-of-the-art methods as well summarized in Table 6. Fifth, we provide a new discussion section, Sect. 5, where we discuss the scope, practical considerations, and limitations of our work. Finally, we update and improve the description of related works in Sect. 6.

2 Background and Datasets

We describe the datasets that we use and provide the background of tensor decomposition here.

2.1 Datasets

We apply our method on five forums in our archive, which consists of (a) three security forums, (b) one online gaming forum, and (c) a group of GitHub repositories of malware software and their authors. All of these forums are in English language. We provide a brief description of these datasets below.

a. Security Forum Datasets The datasets of the security forums are collected from Offensive Community (OC), Ethical Hacker (EH), and Hack This Site (HTS) forums [23]. We use our own python script to crawl and clean the data. They span 5 years from 2013 to 2017. Users in these forums initiate security-related discussion threads in which other interested users can post to share their opinion. The threads fall in the grey area, mainly discussing both “black-hat” and “white-hat” skills. “Black-hat” hacking skills are applied with malicious intention whereas “white-hat” skills with benign intention. A brief description of these forums is presented below.

- (i) **OffensiveCommunity (OC):** As the name suggests, this forum contains “offensive security” related threads, namely, breaking into systems, selling hacking services etc. Posts in this forum consist of step by step instructions on how to compromise systems, and advertise hacking tools and services.
- (ii) **HackThisSite (HTS):** As the name suggests, this forum has also an attacking orientation. There are threads that explain how to break into websites and systems, but there are also more general discussions on cyber-security.
- (iii) **EthicalHackers (EH):** EH forum seems to compose mostly of “white-hat” hackers, as its name suggests. However, there are many threads with malicious intentions in this forum as well.

b. Gaming Forum Dataset We consider an online gaming forum, Multi-Player Gaming and Hacking Cheats (MPGH) [23]. MPGH is one of the largest online gaming communities with millions of discussions regarding different insider tricks, cheats, strategy, and group formation for different online games. The dataset was collected for 2018 and contains 100K comments from 37K users [26].

For completeness, we start with some terminology that are related to the security and gaming forums. Each *thread* has a *title* and is started by its first post, and we refer to subsequent posts as *comments*. The *duration* of a thread is defined by the time difference between the first and last post of that thread. The *active days* for a forum are the number of days when the dataset contains at least one post. A user may leave more than one comment for an article, which leads us to define the *engagement* of a user for that article. An *engagement* has a time duration and intensity in terms of number of comments. Each tuple in our datasets maintain the following format, $F := (\text{forum ID}, \text{thread ID}, \text{username}, \text{date}, \text{post ID}, \text{and post content})$.

c. GitHub Dataset GitHub platform offers the users to create software repositories to store, share, and collaborate on projects and provides many social-network-type functionalities.

We define some basic terminology here. We use the term *author* to describe a GitHub user who has created at least one repository. A *malware repository* contains malicious software and a *malware author* owns at least one such repository. Apart from creating a repository, users of GitHub can perform different *types of actions* including *forking*, *commenting*, and *contributing* to other repositories. *Forking* means creating a clone of another repository. A forked repository is sometimes merged back with the original parent repository, and we call this a *contribution*. Users can also *comment* by providing suggestions and feedback to other authors' repositories. Each tuple in the dataset is represented in the following format, $F := (\text{forum ID}, \text{malware repository ID}, \text{malware author}, \text{date}, \text{action type}, \text{repository content})$.

We use a dataset of 7389 malware authors and their related 8644 malware repositories, which were identified by prior work [29]. This is arguably the largest malware archive of its kind with repositories spanning roughly 11 years. The basic statistics of the datasets are shown in Table 1.

Table 1 Basic statistics of our datasets

| Dataset | Users | Threads/Repositories | Posts | Active days |
|---------------------|-------|----------------------|--------|-------------|
| Offensive community | 5412 | 3214 | 23918 | 1239 |
| Ethical hacker | 5482 | 3290 | 22434 | 1175 |
| Hack this site | 2970 | 2740 | 20116 | 982 |
| MPGH | 37001 | 49343 | 100001 | 289 |
| GitHub | 7389 | 8644 | – | 2225 |

2.2 Background

We leverage the tensor decomposition in our work. We provide the fundamental concepts of tensor decomposition below.

Tensors and Decomposition A d -mode tensor is a d -way array (here we use $d = 3$) [18]. So, we call $I \times J \times K$ tensor a “3-mode” tensor where “modes” are the fixed number of indices to index the tensor; for us the “modes” being the user (U), thread (T), and weekly discretized time (W). Each 3D element of the tensor, $X(i, j, k)$, captures the total number of interaction (in terms of #comments) of user i in thread j at discretized week k or zero in the absence of such interaction. In a decomposition, we decompose a tensor into R rank-one components, where R is the rank of the tensor, as shown in Fig. 2. That means tensor is factorized into a sum of rank-one tensors i.e. a sum of outer products of three vectors (for three modes):

$$X \approx \sum_{r=1}^{r=R} U(:, r) \circ T(:, r) \circ W(:, r)$$

where $U \in \mathbf{R}^{I \times R}$, $T \in \mathbf{R}^{J \times R}$, $W \in \mathbf{R}^{k \times R}$ and the outer product is derived by:

$(U(:, r) \circ T(:, r) \circ W(:, r))(i, j, k) = U(i, r)T(j, r)W(k, r)$ for all i, j, k . Each component represents a latent pattern in the data, and we refer to it as **cluster**. For example, one such cluster in OC represents a group of 29 users that are active in the first weekends of July 2016 and discuss “multi-factor authentication failure” in a group of 72 threads. Each cluster is defined by three vectors, one for each dimension, which show the “participation strength” of each element for that cluster. Typically, one considers a threshold to filter out elements that do not exhibit significant “participation strength”, as we discuss later.

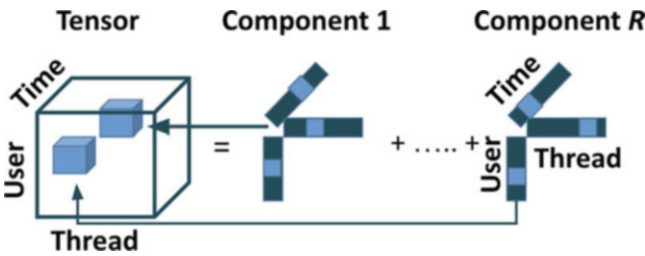


Fig. 2 Visualization of tensor decomposition

3 Our Approach

We present, TenFor, a tensor-based multi-step approach, that identifies events and activities in an unsupervised way. Figure 3 provides the architecture of the platform. The Control module communicates with Interface and Database modules. The algorithmic core is provided by the Tensor Decomposition, Content Profiling, Behavioral Profiling, and Investigation modules.

We present an overview of TenFor, which works in three steps: (a) clustering via tensor-based decomposition, (b) cluster profiling, and (c) cluster investigation as shown in Fig. 3.

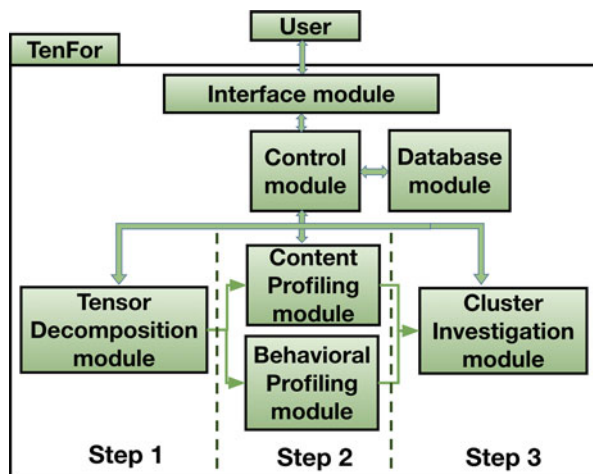
Automated Operation with Optional User Control A key design principle of our approach is to operate parameter-free, and at the same time, provide the end-users to tune these parameters for obtaining results of their interests. Naturally, a savvy end-user can exert even more control by specifying algorithmic parameters with well-defined APIs, especially in the tensor decomposition, which we discuss below. We revisit these parameters at the end of this section.

3.1 Step 1: Tensor-Based Clustering

As a first step, we apply decomposition algorithm on the given input tensor to find the clusters. We provide a quick overview of the challenges and algorithmic choices in the decomposition algorithm below.

Specifics of Tensor Decomposition to Find the Desired Clusters We need to address the following challenges to make the tensor decomposition work well in our domain.

Fig. 3 Overview of the TenFor approach and its steps: Step 1: **Cluster**; Step 2: **Profile**; Step 3: **Investigate**



a. How Can We Decompose the Tensor? Basically, given the input tensor, we use the Canonical Polyadic or CANDECOMP/ PARAFAC (CP) decomposition to factorize the input tensor and find the clusters. But we tailor the CP decomposition algorithm to our needs. For example, the factorization may contain negative numbers in the decomposed components whereas our strategy of capturing the interaction between users and threads at different times is inherently non-negative. We can achieve the non-negative factorization by adding the non-negative constraint in CP decomposition.

b. What is the Ideal Number of Components to Target in the Decomposition? To answer this question, we use the AutoTen method [24] and find the rank (R) of the tensor, which points to the ideal number of clusters to be decomposed into. AutoTen attempts to identify the solution that extracts a large-enough number of components while maintaining a high core consistency, which is a metric for model appropriateness/goodness.

c. How Can We Strike a Balance on Cluster Size? Each cluster is defined by three vectors (user, thread, and time), whose lengths are equal to the dimensions of the tensor as shown in Fig. 2. We need a threshold to determine “significant participation in the cluster”, which is a common practice for (a) avoiding unreasonably dense clusters [31], (b) enhancing interpretability, and (c) suppressing noise. So, the challenge is to impose this sparsity constraint and eliminate the need for ad-hoc thresholding to find the clusters with only significant users, threads, and times. Our solution is to add L_1 norm regularization with non-negative CP decomposition. L_1 regularization pushes the small non-zero values towards zero. Therefore, for each vector, we filter out the zero-valued elements and produce clusters with significant users, threads, and weeks only. In this way, we can eliminate the noisy users, threads, and weeks having the least significant contributions in the forum. The final model that we use for finding the clusters looks like this:

$$\min_{U \geq 0, T \geq 0, W \geq 0} \|X - D\|_F^2 + \lambda (\sum_{i,r} |U(i, r)| + \sum_{j,r} |T(j, r)| + \sum_{k,r} |W(k, r)|)$$

where λ is the sparsity regularizer penalty and $D = \sum_r U(:, r) \circ T(:, r) \circ W(:, r)$.

To find the clusters, we solve the above equation. Since the equation is highly non-convex in nature, we use the well-established Alternating Least Squares (ALS) optimizer as the solver. The effect of λ on the performance of TenFor is described later. An example of a cluster after filtering is shown in Fig. 4 and is further discussed in Sect. 4.

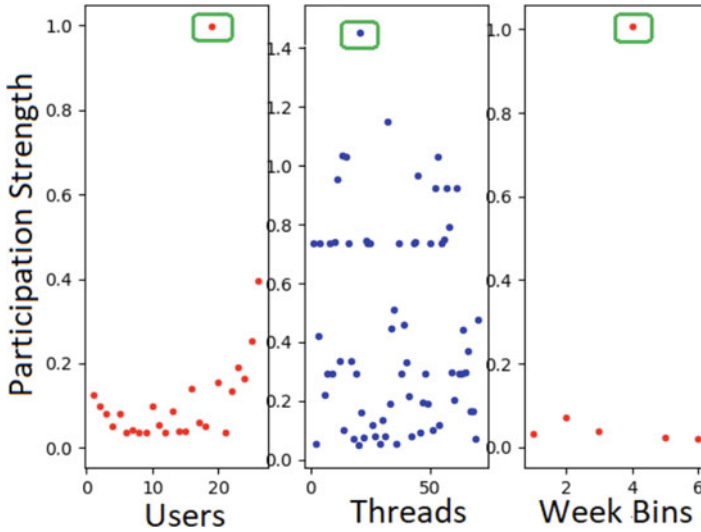


Fig. 4 An example of a cluster (28 Users, 70 Threads, 6 Weeks) from OC. The intensity in each vector helps us identify users, threads and time intervals that are “important” for the cluster

3.2 Step 2: Profiling the Clusters

Having obtained the clusters, we propose to use content-based and behavior-based profiling to provide information and context for each cluster.

Step 2.1. Content-Based Profiling We propose to profile clusters using content with the aid of two interconnected steps.

- a. Cluster characterization:** We identify the top N keywords using TF-IDF from the first post of each thread in each cluster. Prior work argues that the first post of a thread captures the focus of the thread [10]. In the future, we will consider the whole threads. We use the term *cluster keywords* to refer to this set of words. These keywords can already provide a feel for the nature of the cluster, but we also use more sophisticated techniques in the next step.
- b. Cluster labeling:** We give the end-user the ability to define classes of interest that we then use to label the clusters. For ease of use, the end-users can define a class by providing a bag of words. To label the clusters, we compare these bags of words with the *cluster keywords* from the previous step.

To demonstrate this capability, we start with a group of classes that would be of interest to a security analyst. Specifically, we adopt the following classes of interest from prior work [10], which defines four types of threads: **Announcement type (A)** where people announce news and events, including hacking achievements and cyber-attacks; **Product type (P)**, where people buy or sell hacking services and tools; **Tutorial type (T)**, where users post tutorials on how to secure or hack into

systems; and, **General Discussion type (G)**, which is the category for all threads not in the above categories. Again note that although the default classes are A, T, P and G types, the end-users can define his/her own classes other than the above-mentioned fours and respective bags of words for the mandatory labeling.

We then calculate how “relevant” each cluster is to each class type. For consistency, we have adopted the same definitions for these categories as the aforementioned work. To do this, we compute the Jaccard Similarity between the *cluster keywords* and the keywords that define each class type. We label the cluster as *A, T, P, G* type based on the highest Jaccard Similarity score. A cluster can be labeled as **Mix** type if the similarity scores of different types are within a close range (defined as ± 0.02).

Step 2.2. Behavior Profiling To provide more information per cluster, we use behavioral properties, which capture how users and threads interact with each other over time. We provide the following groups of capabilities and plots to the end-users:

- a. **Basic properties plots** of metrics of the clusters in a forum, such as the distribution of number of users, number of threads, number of active days, number of post per thread vs number of thread etc. per cluster of the forum.
- b. **Scree plots** of metrics of clusters, which capture the pair-wise relationships of different metrics of clusters, such as number of threads vs number of users, percentage of active days vs duration (defined as the time difference between the last and the first post of the cluster) for each cluster of the forum as shown in Fig. 7a,b.
- c. **Heat map visualizations** of the clusters and the relative strength of their behavioral metrics. Currently, we use ten behavioral metrics that include the average (over the cluster): average post length per user, number of threads initiated per user, comment to thread participation ratio of the users, number of comments per user, number of active days of the threads etc. In the future, we expand this profiling capability by including more behavioral as well as activity level metrics. We normalize the values of the averaged metrics and present the behavioral profiles using a heat-map-style plot as we show, and discuss later, in Fig. 7c.

The visual depiction helps an analyst to quickly gauge the behavioral profile of the clusters and spot differences. Also, we expand this functionality by developing an automated capability to report the anomalous cluster/s using standard DBScan anomaly detection algorithm [34] in these profiles. We discuss the findings in Sect. 4.

3.3 Step 3: Investigation of Clusters

We develop a suite of capabilities that can help automate an in-depth investigation of the clusters coming from the previous steps. Although this can be done manually, the goal is to make the life of an analyst easier. Our platform provides the user with well-organized and easily accessible information trying to strike the balance between being informative and intuitively interpretable. Moreover, we develop two ways so that the end-users can summarize the clusters: (i) StoryLine View, and (ii) Table View.

Step 3.1. Creating the StoryLine View We develop a systematic and, arguably, more interpretable method to capture the essence of a cluster by highlighting the k most indicative threads in a non-decreasing temporal order as shown in Fig. 1. To accomplish this, we follow the process described below.

Identifying the important threads for the cluster is calculated in the following stages. In stage one, we find an extended list of topics, T_{ext} , for the whole cluster. To do this, we use the commonly-used LDA Bag-of-Words model [33], and we focus on the *titles* of the threads in the cluster threads because the *titles* provide a compact and meaningful summary of the threads. In stage two, we calculate the *relevance scores* of each thread with respect to each topic $t \in T_{ext}$. We associate each thread with the topic with the highest *relevance score*. In stage three, we find the most representing topics, T_{dom} , of the cluster. To achieve this, we find the distribution of the number of threads per topic in the decreasing order and from there we choose the list of *dominant topics*, T_{dom} , which we define as the minimum number of topics that represent at least “thread threshold”, $Th_{dom}=70\%$ (default) of the threads. In stage four, we identify the top R_t most relevant threads based on their relevance score for each of the dominant topics in T_{dom} . We then present them in a non-decreasing temporal order as shown in Fig. 1. Note that the parameter R_t has a default value of 5, but the end-user can adjust it to her liking.

Here, we focus on the *titles* as we want to have the title of thread “tell the story” in a visceral and intuitive way for the end-users. In the future, we will consider the text of the whole thread to find topics and consider more involved topic extraction methods.

Step 3.2. Creating the Table View We provide an alternative way to view all the clusters in the forum in a way that puts emphasis on key authors and key threads. This Table View can provide compact event summarization and key entities in each cluster. We argue that this may be appealing for a different type of analysis. Table 3 demonstrates the Table View that we provide. In our platform, we have clickable links that one can follow to investigate these entities of interest providing an interactive capability. We now present the generation of the columns of Table 3.

a. Identifying important entities: users, threads, and time intervals. We propose a method to identify the most dominant users, threads, and time periods, where significant activity takes place and populate the columns 4, 5, and 6 in Table 3.

Specifically, we propose to identify the top k entities from each cluster, where $k \geq 1$ with the default being $k=3$. We use the factorized vectors to gauge the “importance” of an entity in a cluster as shown in Fig. 4. The green boxes show the entities with the highest “Participation Strength”. From each of the top k weeks, we also report the most active day in terms of the highest #post made in that week.

Note that the parameter k can be modified by the end-user to adjust to her preference or type of investigation.

- b. Representing the nature of the cluster in Table View:** We present another way to capture the essence of a cluster, which we provide as text in the last column of Table 3. Obviously, there are many ways to achieve this. We opt to report the most dominant topics, T_{dom} , per cluster which is a common practice to represent and interpret events [22, 35]. We have already discussed a method to identify the dominant topics above, which can be provided in the final column in Table 3. Note that in Table 3, we start from the dominant topics, but reconstruct the events within each cluster to provide more context to this paper readers.

The Optionally Tunable Parameters of TenFor TenFor can operate without any user input, but we expose the following parameters to a savvy end-user who wants to finetune the operation. Below, we list these parameters and their default values: (a) temporal granularity: week, (b) size of the cluster keywords $N=50$, (c) cluster labels: A, T, P, G/Mix as defined here, (d) thread threshold for dominant topic $Th_{dom}=70\%$, (e) number of relevant threads in StoryLine View $R_t = 5$, and (f) number of top entities in Table View $k = 3$.

4 Results and Evaluation

We apply our method on four forums in our archive discussed in Sect. 2. We discuss the output of each step of TenFor for three security forums below.

4.1 Step by Step Output Provided by TenFor

Step 1 We do a tensor decomposition for each forum. We provide an overview of the results of our decomposition in Table 2. Note that we opt to use *week* as the unit of time, but we considered with days and months as well. In Fig. 5, we show the effect of the bin size on the average size of the cluster in terms of users, threads and time intervals. Note that using a weekly granularity seems to provide smaller clusters in terms of users and threads, which will likely lead to higher cohesiveness. This further corroborates our intuition. As our goal is to capture events, a week

Table 2 Properties of the clusters in OC, HTS and EH. Here U=user, Th=thread, W=weeks and the percentage of users, threads in a particular type of cluster is based on total number of users, threads in each forum

| Forum | Initial Tensor Size | | | Filtered Entities in clusters | | | # Cluster | | | # Cluster per type | | | Type A (%) | | | Type T (%) | | | Type P (%) | | |
|-------|---------------------|------|-----|-------------------------------|------|-----|-----------|---|---|--------------------|-----|------|------------|------|-----|------------|---|----|------------|----|--|
| | U | Th | W | U | Th | W | A | T | P | Mix | U | Th | U | Th | U | Th | U | Th | U | Th | |
| OC | 5412 | 3214 | 240 | 1086 | 2505 | 107 | 25 | 7 | 5 | 8 | 5.7 | 21.2 | 5.1 | 14.3 | 3.8 | 14.3 | | | | | |
| HTS | 2970 | 2740 | 240 | 196 | 676 | 59 | 12 | 3 | 3 | 3 | 1.5 | 7.9 | 2 | 3.7 | 1.6 | 3.7 | | | | | |
| EH | 5482 | 3290 | 240 | 315 | 424 | 82 | 15 | 3 | 6 | 4 | 1.1 | 2.3 | 2.8 | 2.2 | 0.6 | 1.3 | | | | | |

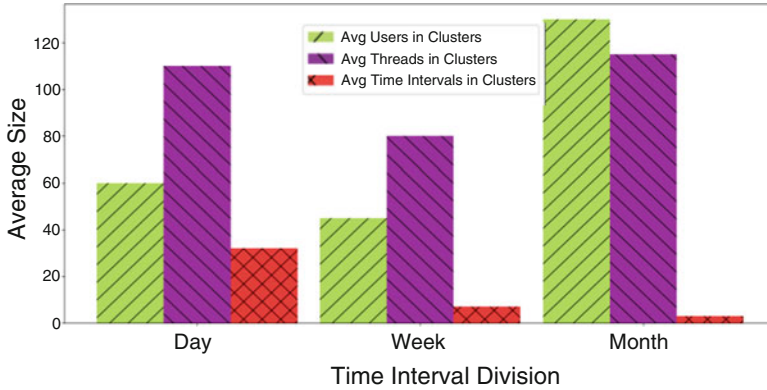


Fig. 5 The effect of the time granularity on the average size of the clusters in terms of (a) users, (b) threads, and (c) time intervals for the OC forum

seems to strike a good balance between a day, and a month, which could be too short and too long respectively.

We find the target number of clusters with the method described earlier. We get a total of 52 clusters from all three forums of which 25 clusters from OC, 12 clusters from HTS, and 15 clusters from EH. Note that we did experiment with more clusters than the ideal number, but that yielded extremely small clusters (e.g. 2 users, 3 threads, 1 week).

Step 2a. Content-Based Profiling and Labeling We use the A, T, P, or Mix/G labels, which we defined earlier. We set the # cluster keywords, $N=50$. Note that we report Mix and G types together here for the ease of presentation.

An overview of the clusters and their properties for all three forums is presented in Table 2. Specifically, we find the following distribution of clusters: (a) **26% of the clusters correspond to real security events**, such as attacks, (b) **22% of them represent black market communities** for malware tools and services, and (c) **32% of them represent security tutorials, events, and communities**, with most tutorials sharing malware and penetration techniques.

Step 2b. Behavior Profiling We provide the functionality to profile clusters based on their activity and dynamics. Apart from providing a general understanding, the analysis can help us spot outliers, which the end-users can investigate in Step 3.

First, our TenFor platform provides some plots of basic properties as described earlier. Figure 6 shows one basic property plot which is number of post per thread vs number of thread for each cluster in OC forum. This figure answers the question: *how do users engage in each type of cluster?*. For example, we find that Announcement type clusters have a large number of posts per thread. Upon close inspection, we also find many self-comments, as the thread initiator clarifies issues

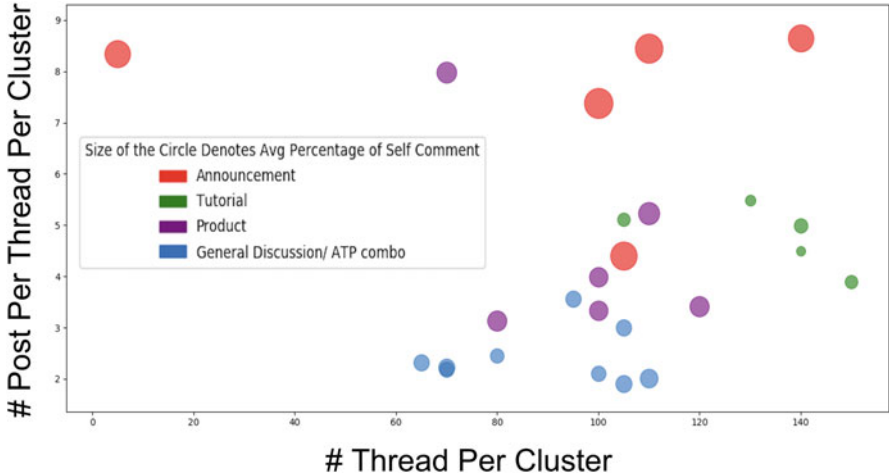


Fig. 6 The average number of posts per thread versus the number of threads per cluster for the OC forum with each type of cluster shown with a different color. Announcement type clusters (in red) exhibit high average posts per thread, while Tutorial type clusters (in green) have high average number of threads per cluster

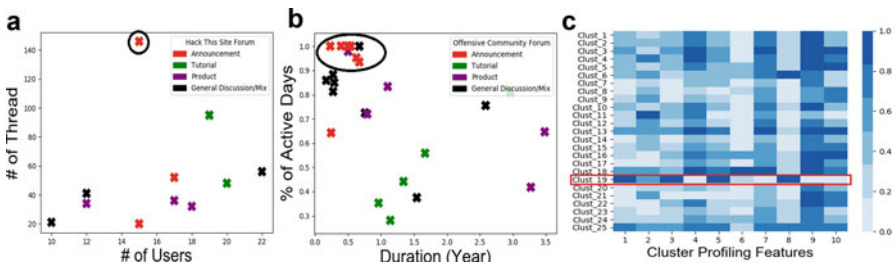


Fig. 7 Behavioral profiling of the clusters. (a) Scree plots of #users vs #threads in A, T, P, Mix/G type clusters of HTS. (b) Scree plots of % of Active Days vs Duration in A, T, P, Mix/G type clusters of OC. (c) Behavioral profiling of the clusters from OC: x-axis is different behavioral features, and y-axis is cluster IDs. Case-study: Cluster 19 has a unique feature intensity profile

regarding her announcement. We also find that Tutorial type clusters contain the high number of threads with only a few self-comments.

Second, we propose to identify clusters of interest by using scree plots which capture the correlation between different properties of these clusters. We show two indicative scree plots in Fig. 7a,b. In Fig. 7a for HTS, the black-circled cluster at the top is an A cluster, where just 15 people participate in a comparatively huge number of 145 threads. Upon further inspection, they are a group of hackers boasting about their hacking success. Some indicative *cluster keywords* of this cluster are *hack, brag, success, breach* etc. (provided by TenFor in wordcloud format as well) which actually substantiate our claim.

Similarly, in Fig. 7b for the OC forum, the encircled clusters exhibit continuous activity: the Percentage of Active Days over the Duration of the cluster is more than 90%! This is an indication of an “urgency” in the cluster when compared with the typically lower Percentage of Active Days. This urgency is amply illustrated by cluster 9 (22 users, 60 threads): users talk about “strike week”, during which the government attacked organized cyber-crime in the second half of March 2015. Strike week created frantic activity in the forum at that time.

Finally, we also provide a compact visual behavioral profile for each cluster shown in Fig. 7c for the OC forum. This can convey condensed information to the end-users visually. For example, cluster 19 (40 users, 88 threads), highlighted with the red box, seems to have a rare combination of active (dark blue) features. Specifically, these features suggest that the cluster exhibits high values of (a) average length of the first post of a thread per user (feature 1), (b) average ratio of #comments to #threads which a user generates or participates in (feature 3), and (c) average #comments per thread (feature 5). This behavior of the cluster is aligned with a *Tutorial* type cluster: (a) the first post is usually long, (b) tutorials often spark discussions, leading to multiple comments by a user in a thread, and (c) there are many questions and “thank you” comments in a tutorial thread. Note that this is also the label that our content-based labeling suggests.

Step 3 We showcase how we can enable a deeper analysis for each cluster with (a) Table View, and (b) StoryLine View. An example of our Table View is presented in Table 3 where we highlight three selected clusters from each forum and we provide the information in terms of the type of the cluster, most significant threads, users, and dates. The final column is populated with the *dominant topics*, though here, we provide a manually-enhanced reconstruction of events for presentation purposes. As explained earlier, we also present a StoryLine View where we identify the top- k most indicative threads of the cluster which provides a human-readable thumbprint of the cluster. In Fig. 1, we show such a result that was generated automatically for cluster 7 (34 users, 125 threads) of the OC forum. We find that the dominant topic, ransomware, represents 81% of the *titles*. With the default settings, TenFor reports the top $k=5$ *titles* per *dominant topic* based on the highest *relevance score* in a sorted timeline fashion. From this StoryLine View, the analyst can easily come into a conclusion that the cluster actually captures the spread of SimpleLocker. Therefore, this view is particularly useful for clusters that capture an event or a discussion, as they can provide the evolution of the event as captured by its most dominant threads.

We discuss 9 of the clusters in Table 3 in more detail to show-case the kind of information that we can gain.

a. Detecting emerging security threats. First, several clusters consist of events that discuss novel security threats. For example, cluster 7 and 12 of OC revealed the growing concern of an extensive outbreak of the SimpleLocker ransomware and the RAT virus respectively. Also, cluster 2 of EH provides a timely warning of the explosive outbreak of Locky ransomware in Feb 2016.

Table 3 Investigating nine clusters identified by TenFor reveals interesting activities. (CID is the id of the cluster)

| Forum-CID | No. users | Type | Top threads | Top users | Top dates | Events and explanation |
|-----------|-----------|------|-------------------|--|--------------------------------------|---|
| OC-7 | 34 | P | 3502, 4843, 4841 | S. Prajapati 23, Cyberseason, Assassin | Dec 2 and 15 2015, Feb 13 2016 | (a) A market of 34 buyer/sellers of decryption tools against SimpleLocker ransomware with peaks in Dec 2015 and again in Feb 2016, which mirrors the outbreak events of SimpleLocker. |
| OC-8 | 54 | A | 2562, 1228, 1234 | V4nD4L, RF, Pratham | Feb 4, 19, and 28 2016 | A peak is detected: (a) when V4nD4L claimed success in hacking Facebook in Feb, 2016, (b) V4nD4L recruits seven members in a hacking group. |
| OC-12 | 42 | T | 804, 6995, 2099 | Dragunman, Pratham, L1nkm3n | June 4, 9 and 19 2015 | (a) Five people collaborated to spread the RAT virus; (b) Dragunman shared tutorials on hacking into banks; (c) Pratham promoted several YouTube videos on hacking WiFi networks. |
| HTS-3 | 63 | A | 890, 10594, 11349 | TheMindRapist, cdrain, Bhaal | Oct 5 and 12, Nov 22 2013 | TheMindRapist announced a hacking web-platform where people can submit the URLs they want to have hacked in Oct 5, 2013. |
| HTS-6 | 39 | T | 1125, 234, 6788 | Ninjex, Rajor, mShred | Apr 7, Aug 22 and 31 2014 | A peak in activities is observed when Ninjex and mShred shared tutorials for building hacking tools during the reported Top Dates. |
| HTS-12 | 18 | P | 3453, 4467, 8901 | whacker, DoSman, Bhaal | April 10 and 28, May 12 2016 | DoSman offered a 30 days free trial of a DoS attack tool with a peak in April, 2016. |
| EH-2 | 31 | A | 7263, 8762, 9127 | DarkKnight, Don, VandaDGod | Feb 1 and 9 2016, May 15 2017 | DarKnight was a victim of Locky ransomware in Feb 2016, which sparked a large discussion. Also WannaCry ransomware created a huge fuss in security world in May 2017 |
| EH-3 | 26 | P | 4563, 213, 4498 | hayabusa, dynamik, azmatt | May 12 2017, July 3 2017, Dec 2 2017 | The peak at the Top Days is due to hayabusa and azmatt offering to sell malware tools: xchat tool for windows, hidden surveillance tools, webcam hacking tool etc. |
| EH-6 | 46 | T | 1251, 8325, 8338 | Don, D3vil, VandaDGod | Nov 19 and 24 2017 | VandaDGod, an expert Linux hacker, shared a popular tutorial series of hacking in Kali Linux in Nov 2017. |

b. Identifying bad actors and their tools. Our analysis can lead to important bad actors with Internet-wide reputation. Interestingly, it seems that hackers use their usernames consistently around Internet forums, possibly enjoying their notoriety. For example, our analysis (also shown in Table 3) leads to the usernames of hackers, “V4ND41”, “Dragunman” and “VandaDGod”. A simple Internet search of these usernames quickly leads to people with significant hacking activities and hacking tutorials on YouTube offered by them. Furthermore, we find that “VandaDGod” is active in multiple clusters in EH forum. In July 2019, a hacker group “VandaTheGod” is reportedly accused of defacing dozens of government sites [7].

4.2 Evaluation of TenFor with Real Data

Evaluating the effectiveness of our approach and tool is inherently difficult due to the open-ended and subjective nature of the problem. We list our efforts to assess the precision and recall of our approach by examining the precision and recall to the best of our capabilities.

A. Precision We present the evidence that our clusters are meaningful using several different angles. We find that **83%** of our clusters revolve around interesting events and each cluster shows high intra-cluster thread similarity. This is validated by a group of security experts and further corroborated via crowdsourcing and the REST methodology [10].

- 1. Manual evaluation from domain experts.** We use a group of 3 security researchers to manually investigate all 52 clusters from all three forums. We asked the experts to (a) assign a score (out of 100) for each cluster based on the topic cohesiveness, and (b) summarize the important event(s) in each cluster, if they think the topic cohesiveness score crosses 70. Our experts determined 43 clusters containing 55 significant events based on the majority vote.
- 2. Manual evaluation via crowd-sourcing.** We recruited nine judges among graduate students across campus to check the 52 clusters whether they contain noteworthy events and assign a similar score per cluster like the domain experts. A key difference is that our volunteers make their decisions based on 10 randomly selected *thread titles* from each cluster. For calibrating their sensitivity, the judges were given two sample clusters before the evaluation with (a) randomly selected thread titles, and (b) titles from the same topic. Note that we declare a cluster as cohesive if at least five of the judges assign a topic cohesiveness score ≥ 70 . The group declared 41/52 clusters (79%) as cohesive containing 56 events. For 52 clusters and 9 judges, we calculate the Fleiss’ kappa score [8], $\kappa = 0.699$, which is substantial enough to come to a significant inter-annotator agreement in our context. In Table 4, we provide an overview of the results above. We argue that each combination in Table 4 columns has its own

Table 4 Precision of TenFor: Percentage of clusters declared as interesting and cohesive in our evaluation

| Experts | Crowds | Expert AND Crowd | Expert OR Crowd | REST |
|---------|--------|------------------|-----------------|------|
| 83% | 79% | 71% | 88% | 79% |

merit with the intersection being the most strict and the union being the more inclusive.

- 3. Assessing the cohesiveness.** We corroborate the effectiveness of our content-based labeling (as A, T, P, G type) and assess the cohesiveness of clusters in an indirect way using a state-of-the-art technique, REST [10]. REST follows a thread-centric approach and labels threads along these four categories focusing on the content of a thread. We applied REST for every thread in our clusters. We find that 42 clusters have more than 70% threads of the same type according to REST and they also agree with our cluster label. Note that REST operates at the level of a thread, while we label clusters, which will inevitably introduce “errors”. Thus, we consider the above matching numbers as a good indication for both: (a) the cohesiveness of our clusters, and (b) the accuracy of our labeling approach.

Going one step further, we manually investigated the threads that REST was not confident enough to label. We randomly selected 200 such threads and found that 81% of these threads were aligned with the type of the clusters they were in. Many of these threads were short, and we suspect that REST did not have enough context to assign a label.

B. Recall Quantifying the recall of our approach is even harder. As answering to “*Are we missing important activities and events?*” question is harder to prove, we attempt to argue in favor of our method by providing three types of observations.

- 1. Any spike in activity is caught by TenFor.** We argue that any event that creates significant activity involving threads and users will be caught by TenFor. To provide evidence, we find the top 20 weeks of high activity (in # posts), and the top 50 active users and threads (in # posts) in forum OC. We find that 19 out of the 20 most active weeks, 47 out of the 50 most active users, and 46 out of the 50 most active threads are also identified among the top 5 “performers” in our clusters ($k=5$).
- 2. Several real-life events are caught by TenFor.** TenFor manages to capture several significant data breach events in the clusters from HTS forum including (a) Sony Pictures, (b) Snapchat, and (c) Slack data breach. Users of security forums tend to be more interested in malware and ransomware discussions. For example, among the six most widespread ransomware from 2013–2017 listed in [5], TenFor captured 5 of them in 4 clusters: (a) SimpleLocker(2015–16) event in OC, (b) Locky(2016) and WannaCry(2017) ransomware event in EH, and (c) CryptoLocker(2014) and Petya(2016) ransomware in OC. Therefore, we argue

that significant real-life events which discussed in the forums extensively are captured in the clusters.

C. Comparison with State-of-the-Art Methods We compare TenFor with TimeCrunch [32], which identifies temporal patterns in a dynamic graph. This is the closest state-of-the-art method: our input tensor can be seen as a dynamic bipartite graph. We argue that TenFor is able to find more and meaningful cluster patterns compared to TimeCrunch.

Specifically, applying the default parameter-free setting of TimeCrunch, we find a total of 17 temporal patterns from three security forums, whereas TenFor finds a total of 52 clusters patterns. First, upon further investigation, we find that 13 of these 17 temporal patterns are actually present in our identified clusters. TimeCrunch reports only fixed types of patterns (full/near bipartite core, full/near clique, ranged/constant star etc.) based on Minimum Description Length (MDL) after encoding the model and the output patterns. Encoding larger clusters leads to higher MDL cost, which may be why TimeCrunch reports clusters of smaller sizes. TenFor does not consider any fixed types of pattern types and leverages the power of tensor decomposition. Furthermore, we observe that all 17 clusters are small in size (less than 21 users), compared to the TenFor cluster sizes (as much as 228 users). It seems that TimeCrunch does not identify larger clusters- probably can not “summarize” efficiently and, therefore, does not identify the interesting larger clusters which we show in Table 3.

We also compare TenFor with a widely-used community finding algorithm for Weighted Bipartite Network (CFWBN) [2]. This approach operates on the user-thread space and identifies a total of 771 bipartite communities from all three forums. However, we find 91% of these clusters are small, with ≤ 3 users, and only 35 communities start becoming substantial with ≥ 5 users. We argue that this large number of communities and the absence of time dimension make a follow-up investigation harder for the end-users.

In conclusion, TenFor strikes a balance between reporting too many and too few meaningful clusters compared to previous other methods. Additionally, it provides the end-users with key actors and a timeline of key events in an informative visualization.

D. Generalizability Our approach works equally well on datasets from different types of online forums: Gaming forum and GitHub. We discuss the findings from these forums below.

- (i) **Malware dataset from GitHub.** We apply TenFor on malware dataset from GitHub to show its effectiveness for a different kind of dataset other than security forum. We construct a 3D tensor for GitHub dataset where each element, $T(i, j, k)$, of the input tensor captures the interaction (in terms of the total number of create, fork, comment and contribution performed) between: (a) author i , (b) repository j , (c) per week k . Applying TenFor on this tensor,

we extract a total of 22 clusters. We showcase some of the indicative findings below.

- a. *keylogger is the gateway to malware development.* The biggest cluster (153 authors, 97 repositories, 23 weeks) revolves around developing keylogger malware where we find a very significant engagement of experienced authors. These skilled authors behave like motivating and helping the new developers by issue commenting and contributing in the projects.
 - b. *Windows malware are on the rise.* We find four clusters showcasing windows malware development. These clusters mainly include backdoor, Trojan, bot, and especially ransomware developers. One cluster (33 authors, 40 repositories) is solely devoted to ransomware development during Mar, April 2017 when a surge of ransomware outbreak in real world. The same phenomenon is also observed in security forum as well.
 - c. *Trojan malware for Mac platform is emerging.* We find a small cluster (14 authors, 22 repositories, 3 weeks) engaged in Mac malware development. This bears the notion that Hackers are targeting Mac platform as well.
- (ii) **Gaming Forum.** We wanted to see if our approach would work equally well on different types of online forums of larger size. For this reason, we apply our method on our online gaming forum, MPGH, discussed in Sect. 2. Applying TenFor on this forum, we find 41 clusters with a total of 1.3K users and 3K threads. Apart from finding clusters related to gaming strategy, and tricks for different popular online games, we also find several cyber-crime related activities even in this gaming forum! We highlight the indicative findings below.
- a. *Scamming and cheating are dominant than anticipated.* Interestingly, the biggest cluster with 300 users and 400 threads is focused solely on scamming. The key perpetrators are reported to be Nigerian scammers and a well-known scamming company, “iYogi”.
 - b. *Romance scamming is the new form of scamming.* We identify a sudden emergence of “romance scamming” reports in the mid of August 2018. Apparently, scammers engage in online games, connect with other players, and win their affection and trust, which they use for monetary gain [6].
 - c. *Hacking for hire.* Another surprising behavior is the search for a hacker to exact revenge on a gaming rival, as captured in a cluster with 69 users and 119 threads.

Our initial results hint at a wealth of interesting behaviors in the gaming forums, which we will investigate in the future.

E. Computational Effort The computation required by TenFor is not excessive. The average runtime for preparing the final StoryLine View of the biggest forum with 100K posts, MPGH, takes only 4.35 minutes on average. Our experiments were conducted on a machine with 2.3 GHz Intel Core i5 processor and 16 GB RAM. We use Python v3.6.3 packages to implement all the modules of TenFor. We believe that the runtime can be reduced to seconds if we use more powerful hardware. These

Table 5 Size (in terms of total user and thread) of the top 5 clusters for three security forums with respect to λ . $\lambda = 0.8$ maintains the balance in cluster size

| Clusters | $\lambda = 0.1$ | | | $\lambda = 0.8$ | | | $\lambda = 1.5$ | | |
|-------------|-----------------|-----|-----|-----------------|-----|-----|-----------------|-----|----|
| | OC | HTS | EH | OC | HTS | EH | OC | HTS | EH |
| 1st largest | 541 | 202 | 363 | 160 | 69 | 111 | 57 | 22 | 29 |
| 2nd largest | 444 | 150 | 327 | 151 | 62 | 100 | 43 | 18 | 28 |
| 3rd largest | 399 | 136 | 318 | 106 | 52 | 68 | 37 | 17 | 28 |
| 4th largest | 391 | 131 | 299 | 103 | 41 | 58 | 25 | 17 | 21 |
| 5th largest | 357 | 111 | 268 | 90 | 39 | 56 | 21 | 13 | 19 |

results suggest that TenFor scales reasonably well in practice. The sample code can be found at <https://github.com/RisulIslam/TenFor>.

F. Parameters Tuning λ The only parameter that the end-user needs to know about is Sparsity Regularizer Penalty, λ . Using this parameter, TenFor tries to balance between very large sparse cluster and very small clusters. Setting high value of λ yields very small clusters containing only a few elements that show high participation strength. On the other hand, setting very low value for λ yields very large clusters with lots of elements, which can be hard to dig deeper and extract exact event/s going on. We recommend to use $\lambda = 0.8$ for which the cluster size is balanced and easier to extract the events and understand them visually through StoryLine View and Table View. Table 5 shows the size (in terms of total users and threads only) of the top five largest clusters for the three security forums with respect to the varying values of λ . We find that $\lambda = 0.1$ and $\lambda = 1.5$ result in very large and very small cluster sizes respectively for all three forums whereas $\lambda = 0.8$ maintains the balance. However, we want to clarify that the end-users have the autonomy to change this default $\lambda = 0.8$ upon his/her preference. The sensitivity of the performance of TenFor to λ is more discussed in 4.3 subsection.

4.3 Evaluation of TenFor with Synthetic Data

To evaluate the clustering quality of TenFor, we use a synthetic tensor with flat clusters injected in it as ground truth since there is a lack of such. Also, we compare the performance of CP decomposition against other state-of-the-art decomposition methods. The metrics that we use for the evaluation are **Total Purity (TP)**, and **Rand Index (RI)**. The generation of synthetic tensor, brief description of the evaluation metrics and the comparisons are described below.

Synthetic Tensor, D_Flat, Generation We generate a flat (non-hierarchical) 3-mode tensor for evaluation purpose. The advantage of a synthetic tensor is that they have a well-established ground truth. To stress-test our algorithms, described later,

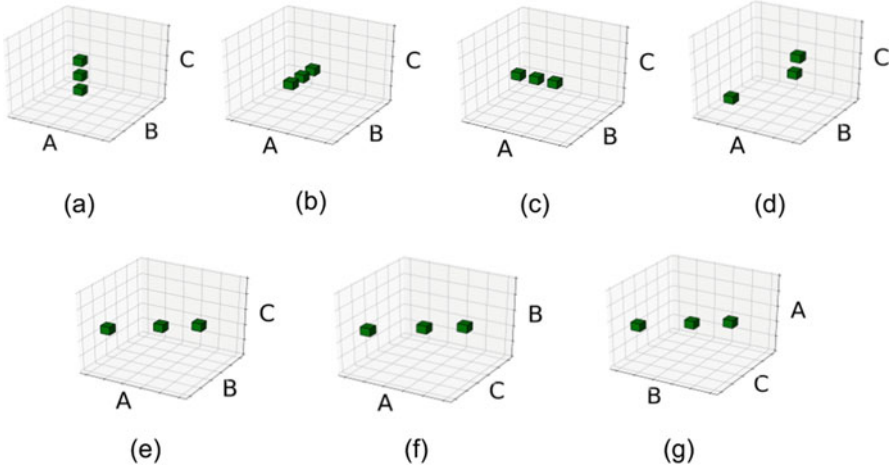


Fig. 8 D_Flat: Creation of challenging (overlapping in 2-modes) clusters in our synthetic tensor by combining the depicted 21 clusters. (a) SSD. (b) SDS. (c) DSS. (d) DDD. (e) DDS. (f) DSD. (g) SDD

we generate a 3-mode synthetic tensor, D_{Flat} . The dimension of D_{Flat} is $300 \times 300 \times 30$, which we find sufficient for our evaluation.

To elaborate, we start from a zero-tensor, Z . Let us consider that Z has three modes A , B , and C , with indices a_i , b_j , and c_k 's along the modes respectively. Then we insert some clusters in Z in such a way that these inserted clusters are not decomposed into further clusters. We call these clusters flat because they span in level 1 only.

Figure 8 shows that a total of 21 clusters (3 clusters from each of the 7 groups) have been introduced which forms the ground truth. Some of the clusters “overlap”, if they get projected in only two dimensions.

The three-letter notation, e.g. SSD, indicates the mode along which the clusters have similar (S) or different (D) values in the corresponding dimension. For example, the three inserted SSD clusters in Fig. 8a have the same a_i s and b_j s, but different c_k s meaning that the three clusters contain same members (across A and B modes) but evolve in different times (along C mode).

How do we insert (i.e. add elements to) each cluster? We identify a center for each cluster and then arrange nodes (equivalently, non-zero elements) around that center by finding the position to insert stochastically. We introduce four parameters to control the size, and other properties of these clusters, which we refer to as **Synthetic Cluster Construction Parameters**. The number of nodes per cluster is controlled by the **concentration** parameter ρ while the **cluster radius**, d , determines the radius of the cluster. The value for each element is drawn from a Gaussian distribution, $G(\mu = 10, \sigma = 3)$.

Evaluation Metrics Evaluating hierarchical multi-modal clustering is challenging as its quality can be analyzed from several different perspectives. For consistency, we adopt metrics from previous methods [20, 28] which we present below.

- (i) **Total Purity.** Total Purity (TP) [20] captures the quality of the clustering and it is measured on a scale of 0 to 1 where $TP=1$ indicates perfect clustering. Intuitively, TP represents the percentage of nodes that are associated with the correct cluster and assumes the existence of ground-truth.
- (ii) **Rand Index.** The Rand Index (RI) [28] is a measure of similarity between two clustering algorithms on the same data. The metric considers all pairs of elements and counts pairs that are assigned in the same or different clusters by each algorithm. RI has a value within [0,1]. A value of 1 represents identical clustering solutions.

Given a set, S , of n elements and two clustering algorithms, X and Y , to compare, the formula to calculate RI is:

$$RI = \frac{a + b}{\binom{n}{2}}$$

where a is the number of pairs of elements in S that are in the same cluster for X and in the same cluster for Y . b is the number of pairs of elements in S that are in the different clusters for X and in the different clusters for Y . In our case, n denotes the total number of non-zero elements in the synthetic tensor.

Evaluation of TenFor Using Synthetic Tensor, D_Flat We evaluate the performance of TenFor in terms of TP and RI in different scenarios. That means we stress-test TenFor by varying the parameters in synthetic tensor construction. Also, we discuss the sensitivity of TenFor to the Sparsity Regularizer Penalty λ . Moreover, we compare the performance of CP decomposition against two other widely-used and state-of-the-art decomposition methods namely Tucker Decomposition (TD) [17] and Dynamic Tensor Decomposition (DynamicT) [36]. For all the above mentioned tasks, we utilize the D_Flat tensor.

- a. **The sensitivity of TenFor to the Sparsity Regularizer Penalty λ .** The Sparsity Regularizer Penalty parameter, λ , is used to select the important cluster members during the decomposition as we explained earlier. Low values of λ create larger clusters, while high values create smaller clusters keeping only the elements with higher participation strength. Clearly, there is a need for a balanced solution that will provide maximal information and insights from the data. Varying the value of the parameter in our study, we find that a value of λ to 0.8 provides the best results with respect to the Total Purity metric ($TP=0.86$). Figure 9 demonstrates the performance of TenFor for varying values of λ .
- b. **The effect of the Synthetic Cluster Construction Parameters: d, ρ, μ, σ .** We also analyze the performance of TenFor by varying radius d , concentration ρ , and data value distribution parameters μ and σ which affect the generation of D_Flat.

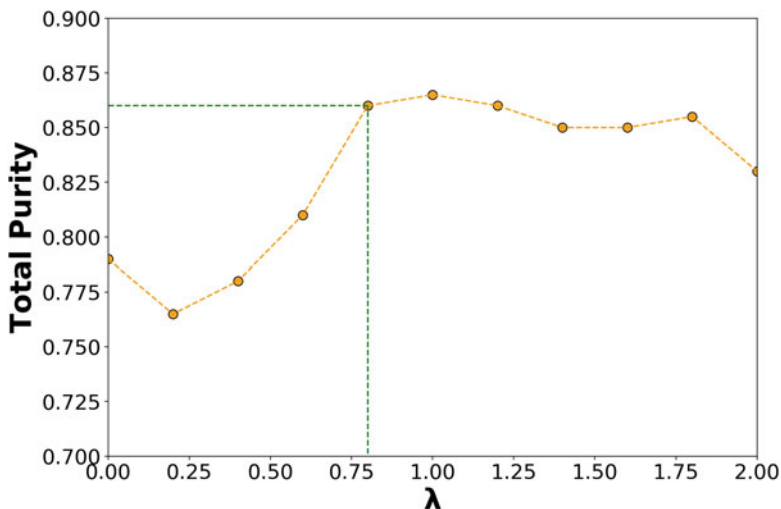


Fig. 9 The effect of Sparsity Regularizer Penalty, λ , on clustering quality metrics TP

We found that TenFor is relatively robust to different values of these parameters. For example, we found a case where doubling the parameter values did not change the performance significantly. Specifically, TenFor shows TP=0.855 for $d = 4$, $\rho = 8/\text{pattern}$, $\mu = 10$, and $\sigma = 3$, and drops to TP=0.83, when we double the parameter values to $d = 8$, $\rho = 16/\text{pattern}$, $\mu = 20$, and $\sigma = 6$.

We intend to evaluate our algorithm more extensively and with more families of synthetic tensors in future.

- c. CP decomposition outperforms other state-of-the-art decomposition methods.** We choose CP model as our decomposition algorithm because it has strong mathematical background, very flexible in adding additional constraints, easy to implement and produce unique clusters. However, we also experiment with two other widely-used and state-of-the-art methods: Tucker Decomposition (TD) and Dynamic Tensor Decomposition (DynamicT) to experimentally verify the effectiveness of choosing CP decomposition. We find that choosing CP decomposition result in better performance (in terms of TP and RI) in finding quality clusters from D_Flat. The reasons that DynamicT and TD do not perform well are obvious. DynamicT is designed for decomposition in dynamic tensors whereas we focus on stationary, pre-constructed general tensor in this work. On the other hand, the drawback of TD is that it does not produce fixed and unique clusters. Table 6 demonstrates the comparison results which demonstrates the reason behind choosing CP decomposition clearly.

The boldfaced numbers in Table 6 suggest that using CP decomposition yields the highest Total Purity (0.856) and Rand Index (0.87) compared to other currently used competitive methods, which indicates that CP decomposition outperforms the existing methods.

Table 6 Performance comparison of TenFor resulted from choosing CP decomposition vs state-of-the-art decomposition methods

| Decomposition algorithms | Total purity | Rand index |
|------------------------------|--------------|-------------|
| Tucker Decomposition | 0.8 | 0.79 |
| Dynamic Tensor Decomposition | 0.81 | 0.8 |
| CP Decomposition | 0.856 | 0.87 |

5 Discussion

We discuss the practical considerations and limitations of our approach.

a. Is our method generalizable to online forums other than security forums?

We argue that TenFor is generalizable to online forums other than security forums. We apply our method on a different types of forum: MPMG gaming forum. Also, we apply TenFor on GitHub malware dataset. In both cases, we find meaningful clusters just like security forums. The findings are discussed briefly in Sect. 4.2. We believe that these findings bears the indication of the generalizability of TenFor.

Going Beyond Online Forums The underlying framework of our approach can be extended to different datasets with more than three dimensions. However, the full functionality of our tool and especially its powerful visualizations with Table View and StoryLine View are meaningful in datasets that have: (a) text data, and (b) temporal dimension. We intend to explore generalizations of our approach to different types of datasets and domains in future work.

b. Is our evaluation sufficient given the absence of extensive ground truth? We

would have loved to have tested our algorithm against an established benchmark. Given its absence, we followed a two-prong approach. First, we evaluate TenFor with synthetic data, where we can know the ground truth, and create a wide range of datasets. We analyzed the effect of the parameter λ and compare the performance of CP decomposition with other two different state-of-the-art methods. Second, we resort to manual evaluation of TenFor on real datasets. We argue that our evaluation provides sufficient evidence of the overall effectiveness and competitiveness of our method. In addition, we will provide our labeled datasets as a building block towards a community-wide benchmark.

c. How can a practitioner use our tool? The interactive UI of our tool makes it very easy for a practitioner to use. First, the practitioner can load his/her data into our tool. The mandatory fields in the input file are: thread ID, username, date, and post content. The end-users can specify which three of the these fields are going to be used for input tensor construction and which text field is going to be used for StoryLine View and Table View construction. Second, the end-user can optionally provide the classes of keywords for cluster labeling purpose although the default classes are A, T, P and G types. Third, all the parameter

fields in TenFor has recommended default values but the practitioner can tune these parameters upon his/her preference. For example, s/he can change the Sparsity Regularizer Penalty λ (default=0.8), number of top entities in Table View (default=3), number of titles per StoryLine View (default=5) and get his/her results of interest.

d. Algorithmic challenges and limitations. We highlight some of the algorithmic challenges of our approach.

- (i) *Number of clusters.* Finding the optimal number of cluster is always a challenging task (NP Hard) for any clustering algorithm. As discussed earlier, we resort to AutoTen tool to do that. AutoTen calculates the rank of the input tensor which helps us determine the appropriate number of clusters. In the future, we could investigate other methods to calculate the rank, and even consider varying the target number of clusters artificially to assess its effect on quality of the resulting clusters.
- (ii) *Cluster size.* Tensor decomposition provides clusters where cluster entities can have very low participation strength. To help the end-users deal with only important entities in each cluster, we leverage the L1 regularization. Varying the λ parameter, the end-user can easily keep only the highly active entities in each cluster although we recommend to use $\lambda = 0.8$ to maintain a balance between too small and too large cluster size. However, the end-users can tune λ to match the needs of the study. In the future, one can elaborate by setting the λ dynamically from the input data.
- (iii) *Cluster labeling.* The default setting of TenFor is to label the clusters as A, T, P, and G/mix classes. However, a practitioner can also determine different types of labels for the clusters depending on the needs of the study. For example, one such labelling could consist on other language or semantic features such as aggressiveness of the language, the sentiment of the users, or specific to keywords that capture topics of interest.
- (iv) *Event representation.* Representing the event is inherently a challenging task. Following the standard practice, we represent the event in a particular cluster by identifying the dominant topics. The Table View shows the events in dominant topic format in the last column. However, we also represent the event in terms of titles in the StoryLine View view. We believe that the topic representation as well as the title representation provides an understandable and complete view of the events going on in a particular cluster.
- (v) *Relating clusters to events.* Although most of the clusters contain single events, our experts find some of the clusters containing multiple events. In that case, understanding the events may become somewhat difficult if the end-users choose low value of “number of titles per cluster” in the StoryLine View. Also, some of the clusters may contain general discussion without any particular topics. But our cluster labeling strategy has the capability to label these types of clusters as G type, which help the end-users understand that those G type clusters are for general discussion.

6 Related Work

Overall, none of the previous efforts combines: (a) using tensor decomposition, and (b) extracting events of interest in an unsupervised manner. The most related work to the best of our knowledge is TimeCrunch [32]. TimeCrunch leverages the MDL principle and is limited to reporting only six fixed types of temporal patterns. It also does not use tensor decomposition and does not include a systematic event extraction mechanism like we do here. We discuss other related works briefly below.

- a. **Mining security forums:** Some recent studies focus on identifying key actors and emerging concerns in security forums using supervised techniques and NLP by utilizing their social and linguistics behavior [21]. Some of these works are empirical studies without developing a systematic methodology. Recent efforts include analyzing the dynamics of the black-market of hacking services [27], extracting malicious IP addresses reported by users in security forums [9]. A recent work [10], REST, identifies and classifies threads given keywords of interest, and we use it to validate our cluster labeling. In this work, we focus on a unique perspective. We mine the important events without any prior knowledge about the forum.
- b. **Mining social networks and other types of forums:** Researchers have studied a wide range of online media such as blogs, commenting platforms, Reddit, Facebook etc. Some recent works analyze the user behavioral patterns observed in Reddit [37] and infer information for the users from their activities on Facebook [3] and GitHub [13, 29, 30] but none of them focus on mining events from the forums. Despite some common algorithmic foundations, we argue that different media and different questions require novel and targeted methods. Event detection is a broad and related type of research [12, 35]. A recent work [22] proposes a hierarchical multi-aspect attention approach for event detection but does not consider the author and temporal dimension as we do here.
- c. **Tensor Decomposition approaches:** Tensor decomposition is a well-studied area of research. For our work, we have used CP decomposition but there are other bunch of tensor decomposition approaches. Tucker Decomposition [17] is the most well-known of them. But the problem with Tucker Decomposition is that it is not capable of generating unique decomposition. There are other tensor clustering approaches but they are applicable only in focused domain, for example, tensor graph clustering to detect higher-order cycles [4], approximation algorithm for 1-d clustering [16]. Another recent tensor-based clustering is Dynamic Tensor Clustering (DynamicT) [36] which works better for dynamic tensors but struggles for general tensors. That means DynamicT is designed for decomposition in dynamic tensors whereas, in this work, we focus on stationary, pre-constructed general tensor. Tensor decomposition has a wide range of applications in diverse domains for categorical data [15, 18, 19, 25]. For example, relatively recently tensor-based techniques have been used in social media analysis. A recent work [1] uses a

two-level tensor decomposition to detect fake news. TimeCrunch [32] focuses on mining some temporal patterns from time-evolving graphs. Although the output of TimeCrunch and TenFor are 3-D clusters, TimeCrunch focus on finding only a few fixed types of patterns like bipartite core, near bipartite core, clique etc. whereas we do not focus on any particular types of pattern. More recent studies [25] use tensor to model multilingual social networks in online immigrant communities. Other works [11, 19] use tensor decomposition to study the online communities and their evolution. But none of these works focus on extracting interesting events from online forums.

7 Conclusion

We propose and develop TenFor, an unsupervised-learning tensor-based approach and tool, to systematically identify important events in a three-dimensional (a) user, (b) thread, and (c) time space. Our approach has the following three main advantages: (a) it operates in an unsupervised way, though the user has the capability to influence its focus, if so desired, (b) it provides visual and intuitive information, and (c) it identifies both the events of interest, and the entities of interest within the event, including threads, users, and time intervals.

Our work is a step towards an automated unsupervised capability, which can allow security analysts and researchers to shift through the wealth of information that exists in online forums in general.

Acknowledgments This work is supported by NSF SATC 2132642 grant.

References

1. Abdali S, Shah N, Papalexakis EE (2020) HiJoD: Semi-supervised multi-aspect detection of misinformation using hierarchical joint decomposition. ECML-PKDD. arXiv preprint arXiv:2005.04310
2. Alzahrani T, Horadam KJ (2016) Community detection in bipartite networks: algorithms and case studies. In: Complex systems and networks. Springer, Berlin, pp 25–50
3. Bayer J, Ellison N, Schoenebeck S, Brady E, Falk EB (2018) Facebook in context: Measuring emotional responses across time and space. *New Media Soc* 20(3):1047–1067, SAGE
4. Benson AR, Gleich DF, Leskovec J (2015) Tensor spectral clustering for partitioning higher-order network structures. In: Proceedings of the 2015 SIAM international conference on data mining. SIAM, Philadelphia, pp 118–126
5. Blog L (2018) Major ransomware events. <https://blog.logsign.com/10-worst-ransomware-attacks-in-the-last-five-years/>
6. Carter C (2019) Romantic scamming in gaming forum. <https://www.stuff.co.nz/auckland/106254141/romantic-scammers-preying-on-players-of-online-game-words-with-friends/>
7. Claster A (2019) News of hacking by vandathegod. <https://www.databreaches.net/dozens-of-government-websites-defaced-by-vandathegod-hacktivists/> [accessed March-2020]

8. Fleiss JL, Cohen J (1973) The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educ Psychol Meas* 33(3):613–619
9. Gharibshah J, Papalexakis EE, Faloutsos M (2018) RIPEX: Extracting malicious ip addresses from security forums using cross-forum learning. In: PAKDD. Springer, Berlin
10. Gharibshah J, Papalexakis EE, Faloutsos M (2020) REST: a thread embedding approach for identifying and classifying user-specified information in security forums. In: ICWSM
11. Gujral E, Papalexakis EE (2018) SMACD: Semi-supervised multi-aspect community detection. In: ICDM. SIAM, Philadelphia, pp 702–710
12. Hasan M, Orgun MA, Schwitter R (2019) Real-time event detection from the twitter data stream using the twitter news+ framework. *Inf Process Manag* 56:1146, Elsevier
13. Islam R, Rokon MOF, Darki A, Faloutsos M (2020a) Hackerscope: the dynamics of a massive hacker online ecosystem. In: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp 361–368
14. Islam R, Rokon MOF, Papalexakis EE, Faloutsos M (2020) Tenfor: a tensor-based tool to extract interesting events from security forums. In: 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp 515–522
15. Islam R, Rokon MOF, Papalexakis EE, Faloutsos M (2021) Recten: a recursive hierarchical low rank tensor factorization method to discover hierarchical patterns in multi-modal data. In: Proceedings of the International AAAI conference on web and social media
16. Jegelka S, Sra S, Banerjee A (2009) Approximation algorithms for tensor clustering. In: International Conference on Algorithmic Learning Theory. Springer, Berlin, pp 368–383
17. Kim YD, Choi S (2007) Nonnegative tucker decomposition. In: 2007 IEEE conference on computer vision and pattern recognition. IEEE, New York, pp 1–8
18. Kolda TG, Bader BW (2009) Tensor decompositions and applications. *SIAM Rev* 51(3):455–500
19. Liu Y, Yan G, Ye J, Li Z (2019) Community evolution based on tensor decomposition. In: ICPCSEE. Springer, Berlin pp 62–75
20. Luu T (2011) Approach to evaluating clustering using classification labelled data. Master's thesis, University of Waterloo, Waterloo
21. Marin E, Shakarian J, Shakarian P (2018) Mining key-hackers on darkweb forums. In: ICDIS. IEEE, New York, pp 73–80
22. Mehta S, Islam MR, Rangwala H, Ramakrishnan N (2019) Event detection using hierarchical multi-aspect attention. In: WWW, pp 3079–3085
23. Online Forums (2021) Ethical Hacker, Hack this site, Offensive Community, MPGH. <https://www.ethicalhacker.net/>, <https://www.hackthissite.org/>, <http://offensivecommunity.net/>, <https://mpgh.net/>
24. Papalexakis EE (2016) Automatic unsupervised tensor mining with quality assessment. In: SDM16. SIAM, Philadelphia, pp 711–719
25. Papalexakis E, Dođruöz AS (2015) Understanding multilingual social networks in online immigrant communities. In: WWW, p 865
26. Pastrana S, Thomas DR, Hutchings A, Clayton R (2018) Crimebb: enabling cybercrime research on underground forums at scale. In: WWW, pp 1845–1854
27. Portnoff RS, Afroz S, Durrett G, Kummerfeld JK, Berg-Kirkpatrick T, McCoy D, Levchenko K, Paxson V (2017) Tools for automated analysis of cybercriminal markets. In: WWW, p 657
28. Rand WM (1971) Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc* 66(336):846–850
29. Rokon MOF, Islam R, Darki A, Papalexakis EE, Faloutsos M (2020) Sourcefinder: finding malware source-code from publicly available repositories in github. In: Proceedings of the 23rd international symposium on research in attacks, intrusions and defenses (RAID). USENIX, New York, pp 149–163
30. Rokon MOF, Yan P, Islam R, Faloutsos M (2021) Repo2vec: a comprehensive embedding approach for determining repository similarity. In: Proceedings of the 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, New York

31. Sapienza A, Bessi A, Ferrara E (2018) Non-negative tensor factorization for human behavioral pattern mining in online games. *Information* 9(3):66, multidisciplinary Digital Publishing Institute
32. Shah N, Koutra D, Zou T, Gallagher B, Faloutsos C (2015) Timecrunch: Interpretable dynamic graph summarization. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1055–1064
33. Sharma D, Kumar B, Chand S (2017) A survey on journey of topic modeling techniques from SVD to deep learning. *IJMECS* 9(7):50, modern Education and Computer Science Press
34. Sheridan K, Puranik TG, Mangortey E, Pinon-Fischer OJ, Kirby M, Mavris DN (2020) An application of dbSCAN clustering for flight anomaly detection during the approach phase. In: *AIAA Scitech 2020 Forum*, p 1851
35. Shi LL, Liu L, Wu Y, Jiang L, Kazim M, Ali H, Panneerselvam J (2019) Human-centric cyber social computing model for hot-event detection and propagation. *IEEE Transactions on CSS* 6(5):1042–1050, iIEEE
36. Sun WW, Li L (2019) Dynamic tensor clustering. *J Am Stat Assoc* 114(528):1894–1907
37. Thukral S, Meisheri H, Kataria T, Agarwal A, Verma I, Chatterjee A, Dey L (2018) Analyzing behavioral trends in community driven discussion platforms like reddit. In: *ASONAM. IEEE*, New York, pp 662–669

Profile Fusion in Social Networks: A Data-Driven Approach



Youcef Benkhedda, Faical Azouaou, and Sofiane Abbar

Abstract User matching across various social networks has received a significant attention in the recent years. Several approaches have been evaluated including discrete user attributes, text mining, network analysis, and more recently machine learning. However, there is a lack of publicly available labeled datasets for this task. Our contribution is twofold. Firstly, we create an open-source framework that collects profiles from various social networks and identifies the true pairs of accounts corresponding to the same user by leveraging user attributes and computer vision. We present a case study dataset that encompasses more than 27k anonymized profile pairs from Quora and Twitter with their corresponding content: 33M tweets and 1.1M Quora answers. Secondly, we evaluate different user linkage schemes and text representation models for the identification of users across these two social networks and discuss the limitations of each approach. Our experiments show that users can be identified with up to 84% accuracy when they have a sufficient amount of generated content in their social accounts.

Keywords Profile linkage · User profiling · User generated content

1 Introduction

The Social Networks (SN) era has enabled access to massive amounts of user generated content (UGC). As of July 2019, it is reported that 4.4 billion users are spanning the web, among which 3.4 billion are registered on at least one social

Y. Benkhedda (✉) · F. Azouaou
Laboratoire de Méthodes de Conception de Systèmes, Ecole Nationale Supérieure
d'Informatique, Algiers, Algeria
e-mail: y_benkhedda@esi.dz; f_azouaou@esi.dz

S. Abbar
Qatar Computing Research Institute, HBKU, Doha, Qatar
e-mail: sabbar@hbku.edu.qa

media platform.¹ Many online users are active in different social platforms at the same time, which poses challenges of identifying them across these platforms. This is particularly true in the case of multi-source data fusion, especially during acquisitions and databases integration. For instance, 90% of Twitter users are reported to use Facebook simultaneously.² This has opened up a wide range of opportunities and challenges. On the one hand, industries were prompted to adopt new marketing strategies. On the other hand, researchers were able to re-explore topics such as *expertise retrieval*, *entity resolution* and *entity matching*.

Different User Matching (UM) schemes have been suggested through the years based on different user features, such as personal discrete attributes (PDA), user generated content (UGC) or network relations (NR). Despite the proliferation of research publications on user linkage problem, it is startling that most of the empirical evaluations were anecdotally executed on small and often non-publicly available datasets, which makes experiments and results impossible to reproduce. This raises serious concerns about the reliability of the presented results and has been previously pointed out by Goga et al. [1]. Given the well-known difficulty for collecting linked profile datasets, we present in this paper the detailed open-source architecture of a framework that detects automatically a user's profiles between Quora and Twitter platforms. We open-source our platform for the community so it can be adapted to other social networks. This can help other researchers to implement new user features in the crawling process. We also publish the collected dataset and encourage the community to use it in their research.³

Our first contribution in this paper is the creation of a User Linkage across Social Networks (ULSN) framework that automates the task of collecting ground truth linked users data. The framework architecture is based on a combination of a username similarity technique and a facial recognition algorithm. The user pairs collection is completed through three main phases. First, a list of personal discrete attributes is retrieved from one SN, e.g., usernames from Quora. Next, we query the second social network with different variants of the extracted PDAs and retrieve profile images of candidate users. Finally, we run a facial recognition algorithm on images of all candidate pairs to (i) make sure the images contain faces, and (ii) check the likelihood that the two faces belong to the same person. Once correct pairs are identified, we collect all their generated content from both networks. This way, we allow for an unprecedented gathering of labeled pairs that we can use to evaluate different kinds of UGC based problems. To guarantee the quality of our system output, we conducted a manual checking of the produced pairs. This led to the validation of 80% of the total produced pair, which is very promising for such an unusual user matching approach.

In the second part of this paper, we highlight some key UM results obtained using different language modeling techniques to represent profile textual content, such

¹ <https://datareportal.com/reports/digital-2019-global-digital-overview>.

² <https://www.pewinternet.org/2018/03/01/social-media-use-in-2018/>.

³ Zenodo: <https://zenodo.org/record/3837711#.Xvr1uJZRUI>.

as bag-of-words, generative probabilistic, and deep learning-based models. The hypothesis here is that the same user might be interested in the same set of entities across different social network platforms. We tested old-school representation models that demonstrated their efficiency throughout the years, such as tfidf and LDA, and compared them to more complex and recent models such as Bert. We also investigated the importance of taking into account the temporal aspect when creating user representations and reported some interesting observations. We recap the paper contributions as follows:

- Generation of the largest publicly user profiles linked dataset with more than 27k account pairs.
- Creation of an end-to-end process to generate ground-truth data for the complex task of user matching in social media platforms.
- Evaluation of different matching approaches, combined with different categories of NLP models, we show that with a bag of words and a good classification algorithm we can achieve an accuracy rate of 84%, without using any elimination process or discrete personal data.

The rest of this paper is organized as follows. Section 2 discusses the related works with a focus on the datasets used in different articles. Section 3 gives a detailed explanation of the ULSN framework architecture and how the data gathering was done. Section 4 describes our data validation process and provides an in-depth overview of our generated dataset. Section 5 explains the setup of the matching approaches we tested. Section 6 discusses our experimental UM results. Section 7 presents some application examples that can be exploited through the use of our framework. We conclude the paper with some remarks in Sect. 8.

2 Related Works

Profile linkage is a long-standing problem that has attracted a significant amount of attention in the recent years. In the case of social networks, people have looked at identifying users across different networks using myriad techniques and datasets. Proposed approaches can be grouped into three categories. (i) Personal discrete attributes: this family of approaches relies on user personal attributes such as full name [2], username [3], email address [4] and gender [2] to identify users across many SNs. (ii) User Generated Content: in this category, users are identified by virtue of textual patterns spotted in their generated content. This is based on the assumption that each user has his unique profile footprint. Several features have been tested, such as language representation [5, 6] or temporal and geospatial components [7, 8]. (iii) Network Relation based: the idea here is to use graph topological features, such as followership/friendship relations to identify users accounts [9, 10].

In the following, we provide an empirical analysis of the datasets used by the community to evaluate profile linkage solutions. One striking observation is that none of the used datasets is public. Which makes it impossible for a fair comparison between techniques or for reproducing their results. The social networks that are usually used for evaluations include: Facebook, Twitter, LinkedIn, Foursquare. For instance, Narayanan et al. [10] applied a heuristic on all three matching fields (username, name and location). They defined a threshold score below which the matching pair is considered as false. Zafarani et al. [11] followed a bio inspection approach. They searched for existing URLs within the user’s bio section mentioning external social accounts of that same user. Motoyama et al. [4] used the Facebook email ID and bio section to search for true matching accounts between Facebook and MySpace. However, they could only collect 1385 true matching user pairs.

Many other researchers used third-party web services to collect their ground-truth base of true matching profiles. Services such as Google+ [1, 5, 12], About.me [5, 13] and FriendFinder [1] were used for this task. As an example, Zhang et al. [5] collected in the first phase more than 150K users from Twitter and LinkedIn. However, they could only validate 4779 user pairs as true matching.

In the last stage of the data gathering process, the researcher re-collects, if necessary, all the profiles data that are needed for his experimentation. Profiles collected data can be of PDA type like usernames, real-names and emails [11–13], network-based type like followers, friends and contacts [10, 14, 15], generic content data like user generated content [13] or specific content data like tag-based content [16], location content [6, 15] or spatio-temporal [17].

Most of the approaches discussed above, based their gathering approach on personal features extracted from users’ bio section, or data found on third-party services, where users provide the links to their accounts. Such truth datasets are quite likely to misrepresent the real-word diversity, considering the fact that the collected users have willingly synchronized their profile coordinates on third-party websites. Furthermore, the experimental data is never shared or made public as shown in Table 1. In this paper, we aim to alleviate those issues by building the foundations of a new automated matching approach, joined by manual curation. Our goal is to produce a highly accurate and reliable dataset of linked user profiles.

3 ULSN System Architecture

In this section we describe the system’s architecture used to collect ULSN data. The system is made modular to allow for automated gathering of labeled data in the future. Figure 1 gives an overview of the different modules that constitute our system.

Dictionary Module The starting point is the establishment of a lexical database with the goal of building the largest possible vocabulary to incorporate into the crawling task. In the case of Quora, each question can be labeled with a maximum

Table 1 Review of datasets used in the literature for profile linkage in social networks. Fb: Facebook, Tw: Twitter, Frs: Foursquare, Li: LinkedIn, Flk: Flickr, Yout: Youtube red: reddit, Goo: Google profiles

| Paper | #Pairs | SNs | Publicly available |
|-----------------------|---------|------------------|--------------------|
| Sun et al. [18] | 7109 | Fb, Flk | No |
| Vosough et al. [8] | 5612 | Fb, Tw | No |
| Riederer et al. [19] | 3031 | Tw, Frs | No |
| Kong et al. [15] | 500 | Tw, Frs | No |
| Motoyama et al. [4] | 1385 | Fb, Mys | No |
| Chen et al. [17] | 2579 | Fb, Tw | No |
| Narayanan et al. [10] | 27k | Tw, Flk | No |
| Bennacer et al. [14] | 474 | Tw, Flk | No |
| Li et al. [6] | 18k–28k | Fb, Tw | No |
| Goga et al. [1] | 850–76k | Fb, Tw | No |
| Zhang et al. [5] | 4779 | Tw, Lnk | No |
| Abel et al. [16] | 712 | Fb, Lnk, Tw, Flk | No |
| Perito et al. [12] | 10k | Goo, Ebay | No |
| Zafarani et al. [11] | 100k | Flk, red, Yout | No |

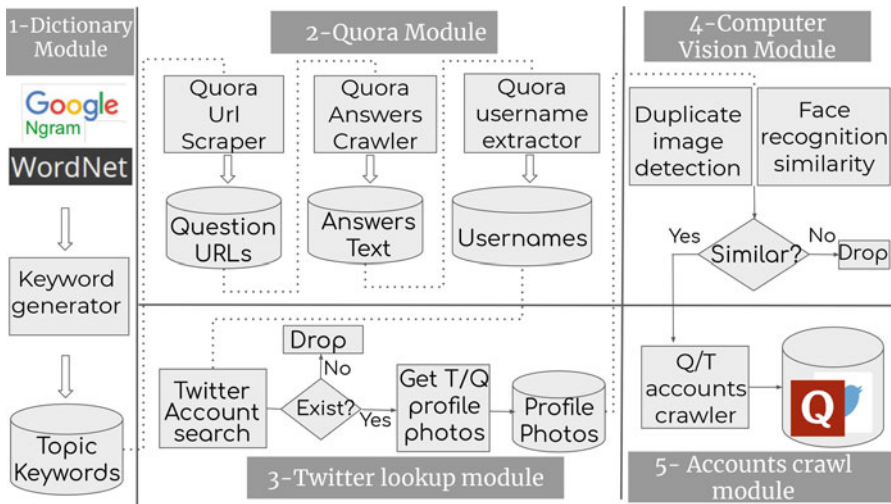


Fig. 1 ULSN system architecture

of 3 topics. Quora has more than 400K discussion topics.⁴ Each Quora topic can be accessed manually through a URL having the following structure: “www.quora.com/topic/topic-name”, however, there is no clear way to automatically retrieve the

⁴ <https://expandedramblings.com/index.php/quora-statistics/>.

topic names. Thus, we employ two English lexical databases, namely: Wordnet and Google-Ngram to guess the maximum number of topic names on Quora.

Quora Crawler Module The aim of the Quora crawler module is to collect all active user profiles from their list of 300M monthly active users.⁵ Most of Quora users have an accessible public profile with an adequate content, i.e., answers on different topical questions. This module first collects all accessible questions of each valid topic identified in the previous step. Then, explores each question to extract all answers along side their authors and timestamps. This module allowed us to retrieve more than 1.7M unique user accounts.

Twitter Look-up Module Twitter look-up module is used to check the existence of Quora usernames in Twitter. Quora uses a simple method for unique username attribution which consists of the user's first name, last name and a sequential number concatenated by a dash character (e.g., july-brown-1234). The username is displayed on the user's profile page. Twitter on the other side, requires from each user to provide their preferred unique username when they create their accounts. It is often the case that Twitter users adapt a username that is a combination of their first and last names. The Twitter look-up module takes each Quora username and creates from it a list of candidate usernames to be looked-up on Twitter. For instance, dashes (-) in Quora usernames are deleted or replaced by underscores, that is, if *july-brown-12345* is a username in Quora, we generate the following set of candidate usernames for Twitter: {*julybrown*, *july_brown*, *brownjuly*, *brown_july*}. The module then issues several http-get requests to check for existence of different variants of the account (e.g., <https://twitter.com/julybrown>). If at least one Twitter username exists, we consider the candidate Quora/Twitter pair as probably matching and keep it for next module use, otherwise the Quora user is dropped from the list.

Computer Vision Module The Twitter look-up module output consists of a candidate username pairs list. Each candidate pair will be examined by the computer vision module. A candidate pair is considered to be a true matching in two cases only: (i) if the profile photos on both social networks are identical, or (ii) if both images contain faces whose similarity confidence is greater than 95% (See Fig. 9). This process led, in our use case, to the collection of 64,759 true matching user accounts (Fig. 2).

Accounts Crawler Module In the final stage of the ULSN collection process, we collect UGCs from both SNs for each true matching pair validated by the vision module. This final step yielded a subset of 34k pairs of user accounts that have at least 10 publicly accessible tweets and 1 Quora answer. Note that this module creates for each accounts pair two files representing respectively their

⁵ <https://foundationinc.co/lab/quora-statistics/>.






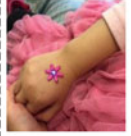




| | | | | | | |
|---------|--|---|---|--|---|--|
| Quora | Ray-Track-4  | Ruby-Sha  | Kada-Hof-6  | Lee-Festiv  | Fed-Down-2  | Solid-Snak  |
| Twitter | | | Kada_Hof  | LeeFestiv  | FedDown  | Solid_Snak  |
| ULSN | Default profile photo detected ✗ | Username does not exist in Twitter ✗ | Faces detected but not similar ✗ | No faces detected and images not duplicates ✗ | Faces detected and similar ✓ | No faces detected but images duplicates ✓ |

Fig. 2 Different matching cases in ULSN. We note that a Quora and Twitter profiles are considered as real matches in two cases only: when the two share the same picture, or the same face

timestamped tweets and Quora answers (one post per line). Files are named by the same sequential number which replaces their usernames. This step is important not to directly disclose the identify of the users in the dataset. Figure 9 presents different scenarios that can occur during ULSN matching process for Quora and Twitter. The input of each scenario is a Quora user identified by his username and profile photo. The output is a decision on whether the candidates pair is true or false. Users having default profile photos, such as Ray-Track-4, or those having usernames with no correspondence in Twitter, such as Ruby-Sha, are all ignored. When the corresponding username account is found on Twitter, we apply the image processing module to check the similarity of the two profile photos. Users having different faces or different profile photos are considered as false matching pairs. This is the case for example of Kada-Hof-6 and Lee-Festiv. Finally, users with similar faces detected (Fed-Down-2), or duplicate images (Solid-snak) are validated by ULSN as true matching pairs.

3.1 *Re-Usability, Code and Implementation Details*

In order to help the research community collect further datasets similar to ULSN, we contribute our code source on Github.⁶ Regarding implementation details, it is important to notice that we use a python module called *twint*⁷ which takes as input a list of Twitter usernames and returns their profiles and tweets. For Quora, we use another python module that uses *Selenium*⁸ which we customized to extract relevant data from HTML pages. It is worth noting that the Quora module might need more customization in the future to adapt to new structure changes of Quora website.

4 Data

We present in this section an overview of the collected dataset. First we show how the validation through human cross-checking was performed, then provide statistics and links to the dataset and the framework, and finally dive into the analysis of the content and temporal characteristic of the data.

4.1 *Data Validation*

To guarantee a high quality dataset, we conducted a manual labeling of the generated ULSN account pairs. The labeling task was assigned to two staff members from our laboratory. The two annotators are asked to go through all detected pairs of users, and label them as being either true or false matching by inspecting the profile photos, usernames and timelines. In addition, annotators were told to label as false all faked or unverified celebrity accounts, such as Albert Einstein or Bruce Lee (Fig. 3). This labeling process led to a validation of 27k user pairs out of 34k initial matching pairs, which represents about 80% of pairs automatically detected by our system. We noticed that rejected user pairs that were labeled as false by both annotators, were mostly cases where users had non-deterministic profile data, i.e., users had one or both of the profile photos with unclear face angle, along with little information found on the profiles timeline (e.g., no bio info and less than 10 tweets and 2 answers). Under similar conditions, the annotators stated that there is not enough information to confirm the matching correctness of the pair.

⁶ Github: <https://github.com/banyous/quora-twitter-scraping>.

⁷ Github: <https://github.com/twintproject/twint>.

⁸ Selenium: <https://selenium-python.readthedocs.io/>.

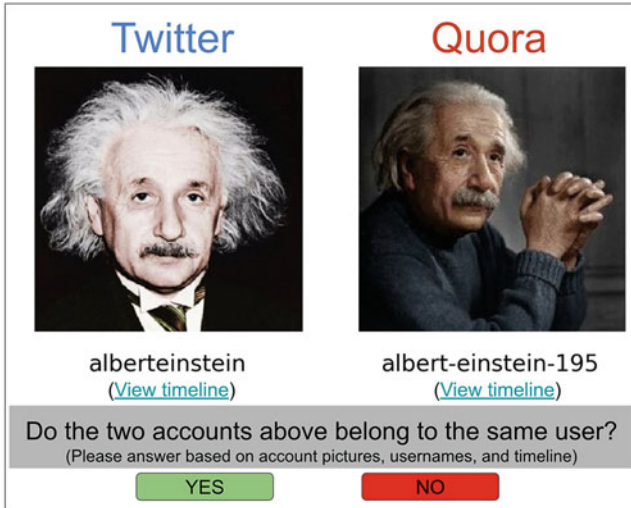


Fig. 3 A screenshot of the labeling tool used by the annotators. The interface displays the Twitter and Quora profile photos, usernames and timeline links

Table 2 General statistics about the dataset. p/u: post per user, med.: median, Avg.: average. Note that the total size of Quora data uncompressed is 1.1 GB vs. 5 GB for Twitter

| | #users | #posts | #avg. p/u | #med. p/u | #unique words |
|---------|--------|--------|-----------|-----------|---------------|
| Quora | 27,049 | 1.08M | 40 | 195 | 1M |
| Twitter | 27,049 | 33M | 1241 | 2841 | 3.2M |

4.2 Dataset Overview

The true matching accounts list produced by the computer vision module consists of 64,759 username pairs from Quora and Twitter. However when we applied the accounts crawler module on this list, we limited the crawling to users having at least 10 tweets and 1 Quora answer (Quora answers are usually much longer than a tweet). Profiles with restricted access content were ignored. The final number of true matching pairs generated by ULSN, and validated through the data validation process, is 27,049. The dataset comes with a total of 33,587,156 tweets (representing 5 GB of uncompressed data) and 1,084,239 answers on Quora (1.1 GB uncompressed). The extracted dictionary of unique words in Quora data consists of 1,012,718 tokens whereas that of Twitter counts 3,212,340 tokens. See Table 2 for more statistics.

The dataset is structured into two folders (/Twitter/, /Quora/) each of which containing 27K files (e.g., /Twitter/1004069.txt, /Quora/1004069-.txt). Each user has one file in each folder with the same 7 digit filename identifier. The file contains posts (tweets or answers) collected from the user profile and ordered by their publication date. Each tweet comes with a

Table 3 Top 10 used languages and extracted locations from text in both Quora and Twitter

| Top locations | | Top languages | |
|---------------|------------|---------------|------------|
| Quora | Twitter | Quora | Twitter |
| China | India | English | English |
| Pakistan | Vietnam | Latin | Spanish |
| USA | London | Hindi | Dutch |
| America | America | French | French |
| Europe | China | German | German |
| Mumbai | Canada | Spanish | Portuguese |
| Australia | Pakistan | Italian | Italian |
| Africa | California | Indonesian | Arabic |
| Japan | Australia | Danish | Indonesian |
| Chennai | Mumbai | Dutsh | Hebrew |

timestamp and the tweet text. Similarly, each Quora answer comes with a date and the answer text. The first line in each Quora/Twitter file represents the bio of the user. It is worth noting that we deliberately deleted from our dataset any field or text referring to a post/author ID for the purpose of keeping the total anonymity of the collected data. The final dataset and the ULSN framework architecture are both accessible through Zenodo⁹ and GitHub¹⁰ platforms respectively.

4.3 Content Analysis

In this section we present a brief content analysis of the dataset. Figure 4 displays the word clouds of the two networks. We note that Twitter content is more social oriented. Sociable terms like love, free and YouTube are quite frequent. In the other side, Quora is more of an educational oriented platform, which explains the prominence of more professional words, such as business, company and marketing. We also analyzed the users employed language, using a simple yet effective python module called LangID¹¹

⁹ Zenodo: <https://zenodo.org/record/3837711#.Xvr1uJZRU-I>.

¹⁰ Github: <https://github.com/banyous/Quora-and-Twitter-crawler-and-user-matcher>.

¹¹ Github: <https://github.com/saffsd/langid.py>. The idea here is that we analyze the users language by extracting the dominant language of each user profile. We then count the frequency of each detected language across the dataset. We employ the same approach by counting frequencies of locations detected in users bio. Table 3 summarizes the top detected languages and locations from the two social networks. Non surprisingly, English is the dominant language on both platforms with more than 99% of Quora posts and 91% of Twitter posts being in this language. Twitter is naturally more language diversified than Quora due to the social character of the former one. An other interesting fact is to see "Hindi" as the third most used language in Quora. This can be explained by the prominence of southern region Asian locations in the Quora top detected locations, such as Pakistan, Mumbai, and Chennai. This has a relation with our starting point: Quora, which is quite prominent in India (Quora stats: <https://www.alexa.com/siteinfo/quora.com>). It is not surprising to see Indian locations frequent in Twitter data as well.



Fig. 4 Quora and Twitter word clouds

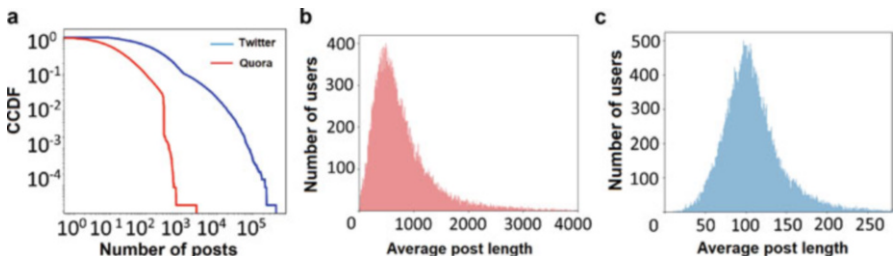


Fig. 5 Activity and content analysis. (a) Posting activity analysis. (b) Answers length. (c) Tweets length

Next we extracted the posting distributions of users in the two SNs. Figure 5a shows the complementary cumulative distribution function in log-log scale. It calculates the probability (y -axis) that a user have at least a certain number of posts (x -axis). As expected, we observe that distributions are skewed in both cases, as very few users make it to have a high number of posts (100 in Quora or 1000 in Twitter). In Fig. 5b,c we present the distributions of posts length measured in number of characters in both SNs. We can obviously see that Quora posts are much longer than those of Twitter, bounded to 140 characters (and 280 characters recently). The difference in profile distributions is also due to fundamental differences in the nature of the two networks.

4.4 Temporal Analysis

In this section, we present some chronological comparisons between the content of the two networks. Figure 6 shows some examples of temporal distributions of the posting volume (tweets and answers on Quora). Figure 6a displays the cumulative number of posts since 2006 for Twitter and 2009 for Quora. We clearly see that there are much more tweets than Quora answers. Figures 6b,c show a sample of

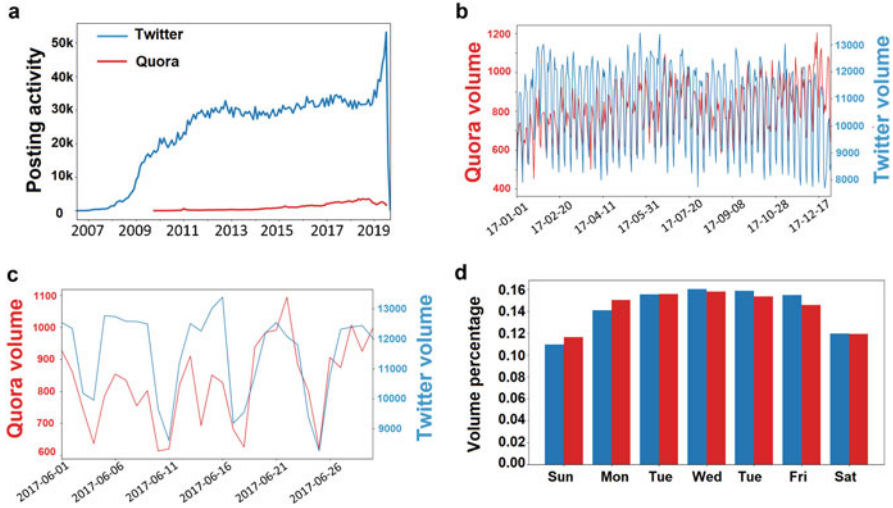


Fig. 6 Temporal features of Quora and Twitter data. posting volume distribution 2017 days, volume distribution of June 2017 days and posting volume distribution per weekday. (a) Cumulative. (b) - 2017. (c) Example month - June 2017. (d) Typical week - 2017

year (2017) and month (June 2017) posting volume respectively. We see a nice correlation between the two series (Quora and Twitter), which is related to the circadian cycle, i.e., nice weekly seasonality, with more posts published towards midweeks. Indeed, There is a clear correlation between the Quora and Twitter posting volumes over time. The two graphs clearly exhibit a drop in the users activity volume in the week-end days and a peak activity in the days preceding the week-end. Figure 6d shows the typical percentage of posts produced each day of the week. It reveals that more posts are expected on Tuesdays, Wednesdays and Thursdays.

5 User Profile Linkage

The first step in content-based user profile linkage is to create representations of users based on the content they generated. For instance, we consider the user profile as the concatenation of all his generated tweets/answers, respectively in Twitter and Quora. Once the representations are established, there are several techniques than can be used to identify the true matching pairs, i.e., those belonging to the same user. In the following, we first formulate the problem of profile linkage in social networks. Then, we describe two approaches that we explored: unsupervised user matching and supervised classification. We finally introduce a clustering-based heuristic to reduce the time complexity of user matching approaches.

5.1 Problem Formulation

Given two social network platforms SN_1 and SN_2 , the user profile linkage problem aims at finding all pairs of accounts $(a_u, a_v) \in SN_1 \times SN_2$ such that $u = v$, i.e. the two accounts $a_u \in SN_1$ and $a_v \in SN_2$ belong to the same user.

Without loss of generality, we assume that each user account a_u in SN_1 has one and only one counter part matching account a_v in SN_2 .

5.2 User Matching Approach

The naive user matching approach performs an exhaustive pair-wise similarity calculation all possible account pairs from SN_1 and SN_2 . The true matching account for each user is the one having the highest similarity score amongst the rest. In other words, the pair of accounts (a_u, a_v) such that $a_u \in SN_1$ and $a_v \in SN_2$ is considered to belong to the same real user if and only if $sim(a_u, a_v) \geq sim(a_x, a_y), \forall (a_x, a_y) \in SN_1 \times SN_2 \wedge (a_u, a_v) \neq (a_x, a_y)$. The highest score is considered when dealing with similarity metrics such as cosine. For distance metrics such as Jaro–Winkler we consider the lowest score. The technique is quite effective when dealing with relatively small size of user datasets as shown in Sect. 6.1. However when the number of users exceeds the range of few thousands, the matching performance decreases significantly due to quadratic calculation time.

5.3 Classification Approach

Here we consider the user profile linkage problem as a binary classification problem. We aim to train a classifier that takes as input two user account representations and outputs a binary result, 1 if the two accounts belong to the same user, 0 otherwise. The labeled data for this supervised learning approach is generated as follows: For each positive example, i.e., true matching pair $(a_u, a_v) \in SN_1 \times SN_2$, we randomly sample N false examples $(a_u, a_y) \in SN_1 \times SN_2 | a_y \neq a_v$. Different values of negative pairs N are tested to study the impact of unbalance ratio on the accuracy of the method.

5.4 Clustering-Based Optimization

The major problem of the user matching approach is the number of comparisons which can be prohibitive. Indeed, for each user account $a_u \in SN_1$, we need to compute similarities with all other accounts $a_v \in SN_2$. One way to tackle this

problem is to pre-process user accounts using unsupervised clustering technique. A good clustering will ensure that similar user accounts end-up as members of the same cluster. At run time, given a user account $a_u \in SN_1$ we first identify the cluster to which it belongs, then compute similarities with only members of that cluster to find the candidate matching account a_v . Note that it is perfectly possible to use different types of representation for the clustering and the matching steps. For instance, one could use LDA topics based vectors to perform the clustering (low dimensionality), then use word2vec or TFIDF based representations to perform the matching.

6 Experiments

In this section, we carry out experiments with the generated data in Sect. 4. We first report the results of user matching using different representations and matching schemes. The second part discusses the impact of user prolificacy and limitations of temporal matching.

6.1 User Matching Results

BERT Our first attempt was to use the modern and powerful pre-trained model BERT [20]. Bert has two main tasks that are next sentence prediction and masked language modeling. The model input size is limited to 512 tokens, which makes predicting raw large documents non-possible. One address this issue we can use text summarization techniques to reduce the size of the user documents to Bert readable format (i.e., 518 or less tokens). However, application of such approach is generally expensive on large corpus. Instead, we adopted the approach presented by [21]. In order to classify user reviews based on Bert model sentence prediction, they tested three simple truncation techniques: Head, Tail and Head+Tail, where they selected respectively the first 518 tokens, last 518 tokens and first 128 + last 372 tokens. They achieved the best classification performance results using H+T truncation and the last layer of the BERT model as the feature embedding model. In our case, the best accuracy results were obtained using the H+T with a simple Bert sentence prediction model. We got a accuracy of 22%. We concluded for now that BERT might not be a good fit for very large text, as in profile linkage where each user is represented with all their contributions (i.e., tweets and/or answers.)

TFIDF Next, we tested more simpler embedding models such as TFIDF, LDA, and doc2vec. Our evaluation has shown that the simple TFIDF outperforms all other models by a large margin. We trained the TFIDF model on the union of all documents (tweets and Quora answers). A startling top-1 accuracy of 61% was obtained using low value of maxdf=10, which is a fairly interesting outcome given

that a random matching would have achieved an accuracy of $0.2\% = 100 \times 1/500$. To reduce the dictionary size we set `mindf=2` to delete noisy terms that occur in only one document in both datasets. Interestingly enough, trying the n-gram range to (1,2) increased the top-1 resulted in a 6% improvement in the accuracy, bringing up to 67%. However, it is important to note that n-grams also increased the size of the vectors from 73K to 692k dimensions (unique tokens), which makes it hard to apply to very large datasets. We also observed that pre-processing text, like removing stop-words and stemming, had a negative impact on results. This can be explained by the fact that limiting the max-features or pre-processing the text leads to the disappearance of rare words (misspellings) that are key to distinguish between users textual footprints.

6.2 Classification Results

To evaluate our results, we compare our matching to the classification matching approach proposed by [6]. They tested 10 different classifiers using linguistic representation models such as `tfidf` and `glove`. We adopt their classification approach using `tfidf`-based weight vectors and use it as a baseline in the paper. We add to this different representation models using LDA and Bert weight vectors. We test different document classification algorithms such as logistic regression and support vector machines. Different negative sampling sizes were tested, with values of N set to 1, 3, 5 and 10. Sampled pairs were constructed through mathematical addition of the two accounts feature vectors. For each value of N , we generate 30 different randomly sampled datasets that are split into training set (80%) and test set (20%). We consider the average accuracy of the 30 iterations as the classifier accuracy. In our case, we are more interested in the positive pairs prediction accuracy, which is the classifier precision or the true positive accuracy (TPA). In Fig. 7, we present the classifiers confusion matrix for $N = 1$. We notice that the accuracy of true positives are relatively low compared to the results obtained with user matching technique, with KNN and RF having the highest TPA of respectively 34% and 30%. Increasing N values decreases significantly the TPA to less than 2% for $N = 10$. This is the result of the classifier predicting all new pairs as negative matching, which is the dominant class in the imbalanced setup (Fig. 8).

Classification techniques under-performed in our context. We believe that the main reason is the lack of language models that accurately captures the representation of long text. Indeed a user in our case is considered as one document that contains thousands of tweets or hundreds of answers, where models such as Bert only capture 512 tokens. Similarly, classification based on LDA rendered low accuracy due weak-representativeness of documents intrinsic features. As for TFIDF, the high sparsity of the user learned vectors had a negative effect on the classifiers learning process.

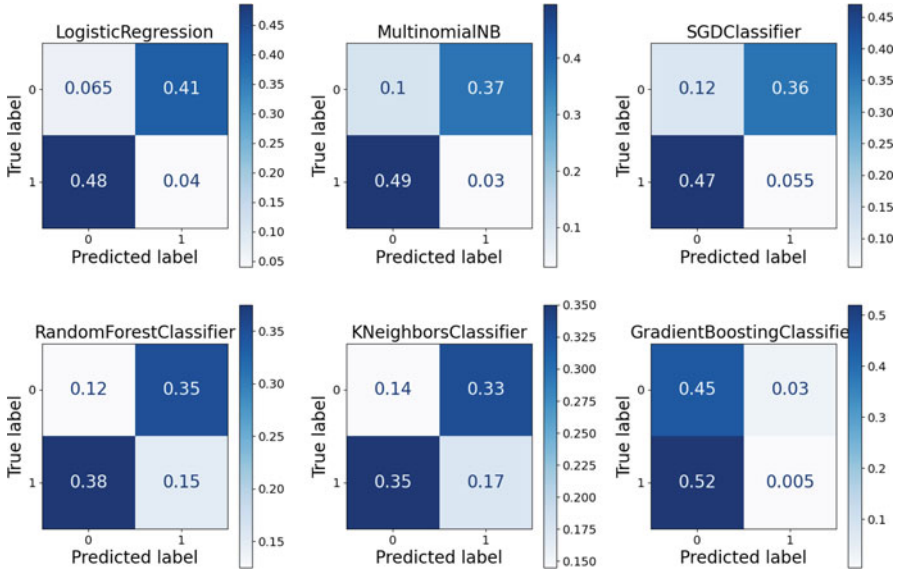


Fig. 7 Percentage accuracy confusion matrix of 6 classification methods applied on S1 sample for ratio of false pairs ($N=1$) and TFIDF vectors

6.3 The Impact of Clustering

We tested different clustering algorithms such as k-means and hierarchical clustering [22] with different values (k) of number of clusters. We used LDA embeddings to compute similarities during the clustering step. We observed that small numbers of clusters lead to better overall linkage accuracy. However the clustering distribution were highly imbalanced. On the other side, increasing the number of clusters reduces significantly the size imbalance among clusters, but affects the overall accuracy of correctly clustered pairs. The best results were obtained for $k = 100$ with sub-space clustering technique which yielded an accuracy of 40%. Using the combination of other embedding techniques such as Bert or Doc2vec produced much lower accuracy.

6.4 The Impact of User Prolificacy

Users in our dataset have different volume of activity in Quora and Twitter. In order to understand how user prolificacy (i.e, volume of produced content) affects the effectiveness of profile linkage, we tested the best performing technique, i.e., TFIDF user matching on four user groups: (a) users with at least 10 tweets and 1 answer, (b) users with at least 100 tweets and 10 answers, (c) users with at least 1000 tweets

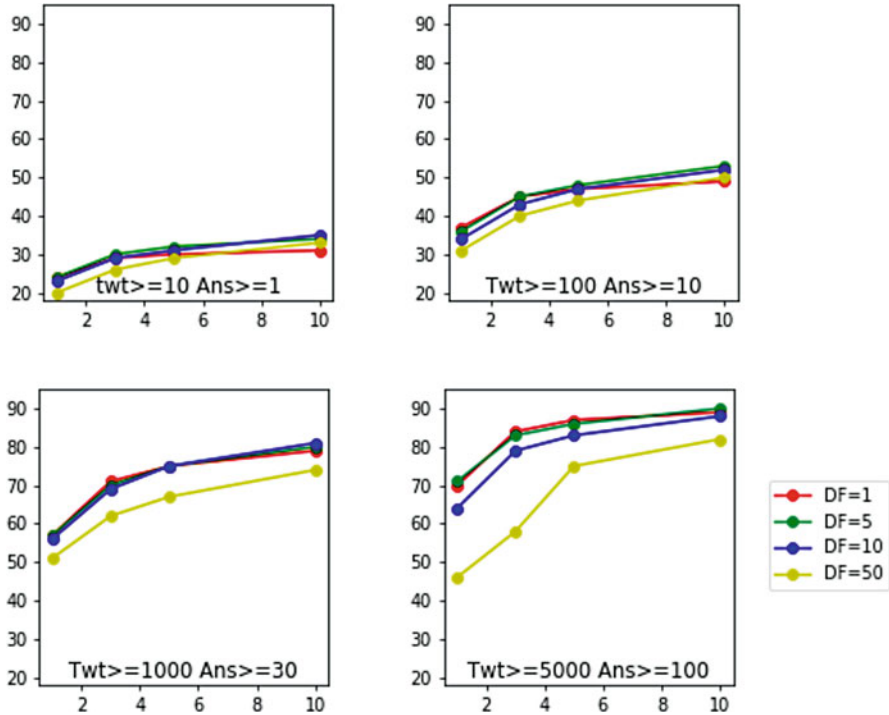


Fig. 8 Different top-k accuracy values for TF-IDF matching with MAXDF varying in range of 1,5,10,50

and 30 answers, and finally (d) users with at least 5000 tweets and 100 answers. The number of users of (a), (b), (c), and (d) groups is respectively 27049, 13064, 1501 and 241 users. Figure 2 shows the top-1, top-3, top-5, and top-10 accuracy results achieved for different values of maxdf (1, 5, 10, 50). One striking results is that the accuracy of profile linkage significantly increases when the amount of available data increases. Indeed, we notice that linking accounts in group (d) can be done with 84% accuracy for maxdf=5. This result is explained by the fact that active users have more available textual features that makes tfidf more efficient in distinguishing between their content. The worst results are observed in group (a).

In order to check the reliability of the previous results, we tested the previous matching algorithm on a greater dataset containing the top active users from (d) merged with normal users from (a), (b), and (c). Using cross-validation, we generated 20 test samples with each sample containing the 239 top active users and 1195 randomly selected users having less than 500 tweets and 10 answers, which makes a rate of 1 active user for each 5 normal user in the dataset. Using this approach we could identify users in group (d) with a top-1 accuracy of 73 %, which is a result close to what we observed earlier when only users from group (d) were considered as candidate matches. This shows that highly active users can still

be highly identifiable even when they are part of a considerably larger group of less active users.

6.5 The Impact of Time

We discuss in this section some ideas that we tried but did not yield to significant improvement on the overall quality of profile linkage task.

First, we tried to use the posting dynamics of users as a filter to reduce the search space. Our intuition was that the Tw/Qu time-series correlation of the same user is more likely to be higher than the time-series correlation of two accounts that belong to two different users.

We tested the correlation relations between each user’s accounts. We perform this task by applying Pearson correlation on the posting frequency of the common time periods between the two accounts. The selected periods are weeks, months and years, and the tests were applied to the four group of users (a), (b), (c), and (d) from section above. As shown in Fig. 9, top average correlation between real users accounts were obtained from yearly frequencies for the four groups. The weekly

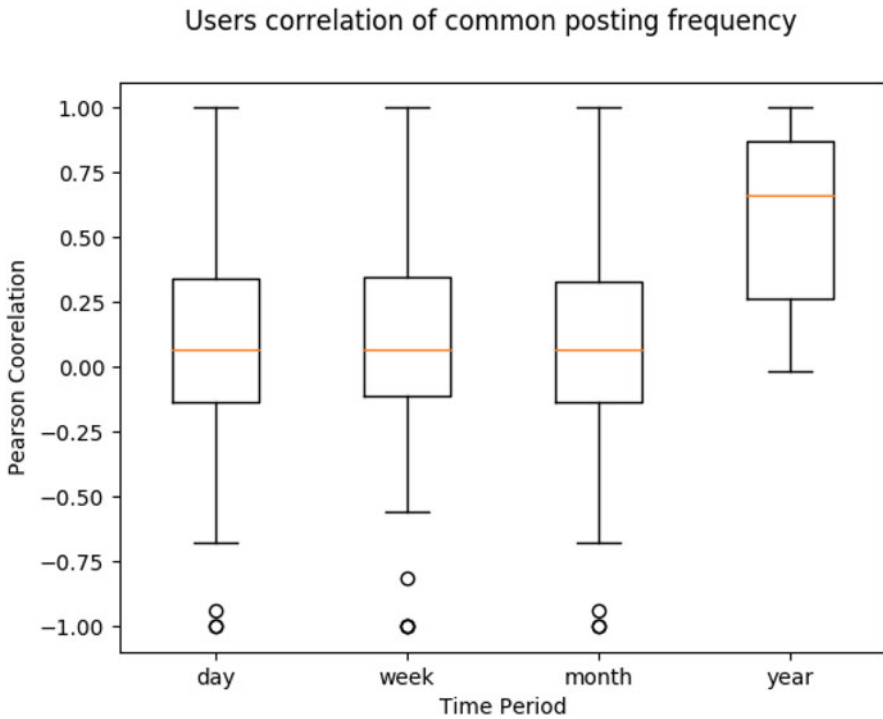


Fig. 9 Periodic Pearson Correlations between Qu and Tw users (S1 users)

and monthly correlations turned to be much less significant, which can be due to the difference in nature between posting behaviours on Twitter and Quora.

We also tried for each pair of accounts $(a_u, a_v) \in SN_1 \times SN_2$ to compare, to build the representation vectors only from documents (tweets and answers) posted in the same time period range. We then take the average accuracy of all the common periods as the matching accuracy score for the pair. We tried several period range units such as same week, same month, and same year. But this did not yield any improvement on the results.

7 Perspectives on Exploiting ULSN Data

A number of applications can benefit from mapping user accounts across different social platforms. Our dataset reveals significant linguistic potential that can trigger a variety of user-related research. We mention in this section the major fields that can benefit from the use of our framework and discuss how our data is relevant to such cases.

7.1 User Matching

ULSN generated data can constitute a base for training, testing and comparing supervised learning techniques that aim at matching users based on their generated content. It will be interesting to investigate the differences and similarities between the textual characteristics of the Q&A and microblog platforms, and how they affect traditional matching schemes. This can include the use of extracted topics, hypertext links and Named Entities. Another interesting aspect is to check the extent to which existing user matching techniques work, when applied to social media platforms that are as different as Twitter and Quora.

7.2 Expertise Retrieval

Expertise retrieval or *Experts finding* refers to the task of finding users with reliable knowledge on specific topics. This task can be explored by considering Quora textual features as the user's expertise reference. Quora offers rich and diversified linguistic structures from which expertise topics can be extracted. A similar approach was previously proposed by Xu et al. [23] in which they evaluated their expertise algorithm by using Quora and Twitter profiles as ground and test sets respectively. However, the dataset they used had 10K users only and was not publicly shared, which makes it impossible for the community to compare with their work. Our public dataset can quickly become the standard in this domain.

7.3 *Authorship Identification*

The task of determining the author of a particular post, based on the analysis of different writing styles, is typically referred to as *Authorship Identification* [24]. More recent studies have been oriented on identifying authors within heterogeneous social networks environment [25]. ULSN will help in exploring this issue by offering highly curated linked profiles between two platforms that are different in many aspects.

8 Conclusion

We suggested in this paper ULSN, a system that maps Quora accounts to their corresponding social accounts. The presented use-case described the process of Quora and Twitter matching that resulted in a dataset of 27k curated pairs of users. The users generated posts consisted of a total of 33M tweets and 1.1M Quora answers. The goal is to make available linked users generated content data, which allows the assessment and re-exploration of user-related applications, such as entity matching and user profiling. We discussed in detail the system architecture and made available its data, so as to let the community use, test, or contribute to the project. In the future, we intend to release other versions of ULSN by taking into considerations more user attributes and by connecting new Social platforms. We can enrich the system output with more demographic and background characteristics, such as gender, race, and expertise. This can be done through advanced computer vision modules or crowd-sourcing platforms.

In the second part of this paper, we presented a preliminary experimental study on the user matching problem using ULSN generated data. Promising results were obtained showing a high precision linkage of user accounts using simple language models such as tfidf with accuracy scores of 67%. Furthermore, based on the analysis we performed on different categories of active users, we found that the more data we have about a given user, the higher the chances to correctly link her two profile accounts. Indeed, for users who posted more than 5000 tweets and 100 Quora answers, we could correctly link their profiles with an accuracy of 84%. Given that our best features are combinations of somewhat unique (rare) words using n-gram, we do believe that the model is quite agnostic to the underlying language. This is different from techniques based on semantics, which may need more repurposing using machine translation to capture the meaning of words across various languages a user may use. This said, we presented the highlights of the top detected language in both Quora and Twitter documents used in our experiments.

One major technical issue we faced when learning the user features is the sparsity of extracted data. To address this problem we used feature selection with different ratio parameters. We also implemented feature selection based on the Chi-Square method. Although not illustrated here, both feature selection methods had a negative

impact on the matching accuracy as they tend to eliminate important features such as misspelled words. Another option we would like to consider in the coming weeks is to adapt dimensionality reduction techniques such as PCA for the textual feature selection problem. Having tested the effectiveness of different language models and pattern analysis techniques, it would be interesting also to explore the directions in which we can use these models at larger scales. External similarity frameworks such as Facebook Faiss can play a critical role in narrowing the matching space with a low computation cost instead of unsupervised clustering.

References

1. Goga O, Loiseau P, Sommer R, Teixeira R, Gummadi KP (2015) On the reliability of profile matching across large online social networks. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, pp 1799–1808
2. Liu S, Wang S, Zhu F, Zhang J, Krishnan R (2014) Hydra: large-scale social identity linkage via heterogeneous behavior modeling. In Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, New York, pp 51–62
3. Tan S, Guan Z, Cai D, Qin X, Bu J, Chen C (2014) Mapping users across networks by manifold alignment on hypergraph. In Twenty-Eighth AAAI conference on artificial intelligence
4. Motoyama, M, Varghese G (2009) I seek you: searching and matching individuals in social networks. In Proceedings of the eleventh international workshop on Web information and data management. ACM, New York, pp 67–75
5. Zhang H, Kan M-Y, Liu Y, Ma S (2014) Online social network profile linkage. In Asia information retrieval symposium. Springer, Berlin, pp 197–208
6. Li Y, Zhang Z, Peng Y, Yin H, Xu Q (2018) Matching user accounts based on user generated content across social networks. *Futur Gener Comput Syst* 83:104–115
7. Goga O, Lei H, Parthasarathi SHK, Friedland G, Sommer R, Teixeira R (2013) Exploiting innocuous activity for correlating users across sites. In Proceedings of the 22nd international conference on World Wide Web. ACM, New York, pp 447–458
8. Vosoughi S, Zhou H, Roy D (2015) Digital stylometry: linking profiles across social networks. In International conference on social informatics. Springer, Berlin, pp 164–177
9. Liu L, Cheung WK, Li X, Liao L (2016) Aligning users across social networks using network embedding. In IJCAI, pp 1774–1780
10. Narayanan A, Shmatikov V (2009) De-anonymizing social networks. arXiv preprint arXiv:0903.3276
11. Zafarani R, Liu H (2013) Connecting users across social media sites: a behavioral-modeling approach. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, pp 41–49
12. Perito D, Castelluccia C, Kaafar MA, Manils P (2011) How unique and traceable are usernames? In International symposium on privacy enhancing technologies symposium. Springer, Berlin, pp 1–17
13. Wang M, Tan Q, Wang X, Shi J (2018) De-anonymizing social networks user via profile similarity. In Proceedings of the 2018 IEEE third international conference on data science in cyberspace (DSC). IEEE, New York, pp 889–895
14. Bennacer N, Jipmo CN, Penta A, Quercini G (2014) Matching user profiles across social networks. In International Conference on Advanced Information Systems Engineering. Springer, New York, pp 424–438

15. Kong X, Zhang J, Yu PS (2013) Inferring anchor links across multiple heterogeneous social networks. In Proceedings of the 22nd ACM international conference on Information and Knowledge Management. ACM, New York, pp 179–188
16. Abel F, Herder E, Houben G-J, Henze N, Krause D (2013) Cross-system user modeling and personalization on the social web. *User Model User-Adap Inter* 23(2–3):169–209
17. Chen W, Yin H, Wang W, Zhao L, Zhou X (2018) Effective and efficient user account linkage across location based social networks. In Proceedings of the 2018 IEEE 34th International Conference on Data Engineering (ICDE). IEEE, New York, pp 1085–1096
18. Sun S, Li Q, Yan P, Zeng DD (2017) Mapping users across social media platforms by integrating text and structure information. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, New York, pp 113–118
19. Riederer C, Kim Y, Chaintreau A, Korula N, Lattanzi S (2016) Linking users across domains with location data: theory and validation. In Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp 707–719
20. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
21. Sun C, Qiu X, Xu Y, Huang X (2019) How to fine-tune Bert for text classification? In China National Conference on Chinese Computational Linguistics. Springer, Berlin, pp 194–206
22. You C, Robinson D, Vidal R (2016) Scalable sparse subspace clustering by orthogonal matching pursuit. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3918–3927
23. Xu Y, Zhou D, Lawless S (2016) Inferring your expertise from twitter: Integrating sentiment and topic relatedness. In 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI). IEEE, New York, pp 121–128
24. Zheng R, Li J, Chen H, Huang Z (2006) A framework for authorship identification of online messages: writing-style features and classification techniques. *J Am Soc Inf Sci Technol* 57(3):378–393
25. Li H, Chen Q, Zhu H, Ma D, Wen H, Shen XS (2017) Privacy leakage via de-anonymization and aggregation in heterogeneous social networks. *IEEE Trans Dependable Secure Comput* 17(2):350–362

RISECURE: Metro Transit Disruptions Detection Using Social Media Mining And Graph Convolution



Omer Zulfiqar, Yi-Chun Chang, Po-Han Chen, Kaiqun Fu, Chang-Tien Lu, David Solnick, and Yanlin Li

Abstract In recent years we have seen an increase in the number of public transit service disruptions due to aging infrastructure, system failures and the regular need for maintenance. With the fleeting growth in the usage of these transit networks there has been an increase in the need for the timely detection of such disruptions. Any types of disruptions in these transit networks can lead to delays which can have major implications on the daily passengers. Most current disruption detection systems did not operate in real-time or lack transit network coverage. The theme of this thesis was to leverage Twitter data to help in earlier detection of service disruptions. This work involves developing a pure Data Mining approach and an approach that uses Graph Neural Networks to identify transit disruption related information in Tweets from a live Twitter stream related to the Washington Metropolitan Area Transit Authority (WMATA) metro system. After developing the two different models, a Dynamic Query Expansion model and a Tweet-GCN to represent the data corpus we performed experiments and comparisons to other existing models, using two different benchmark datasets, to justify the efficacy of our models. After seeing the results across both the Dynamic Query Expansion and the Tweet-GCN, with an average accuracy of approximately 78.9% and 87.3% we were able to conclude that the graph neural model is superior for identifying transit disruptions in a Twitter stream and also outperforms other existing models.

Keywords Data mining · Graph convolution · Dynamic query expansion · Web application · Twitter

These authors contributed equally to this work.

O. Zulfiqar · Y.-C. Chang · P.-H. Chen · K. Fu · C.-T. Lu (✉)

Department of Computer Science, Virginia Tech, Northern Virginia Center, Falls Church, VA, USA

e-mail: omer95@vt.edu; bensochang@vt.edu; pohan@vt.edu; fukaiqun@vt.edu; ctlu@vt.edu

D. Solnick · Y. Li

Washington Metropolitan Area Transit Authority, Washington, DC, USA

e-mail: dsolnick@wmata.com; yli@wmata.com

1 Introduction

Public Transit Networks are an integral part of the infrastructure for all major metropolitan cities. Since they are virtually open to everyone, these transit systems bring in large volumes of daily users or customers. The metro/subway of any city plays an important role by connecting the suburbs and outskirts of the city to the main metropolitan area. This makes them one of the popular modes of transportation for daily commuters. [1]. Back in 2019 the Washington DC Metropolitan Transit Authority reported of having an average daily rail ridership of around 630,000 [2]. That is approximately 315,000 daily riders on the Metro on a given weekday, assuming each rider makes a round trip. Disruptions in service can severely affect these daily commuters and often force them to seek alternative modes of transportation. This could eventually drive customers away, denting the revenue generation for the transit agency.

In today's era of technological advancements, the growing use of social media applications and platforms allows the users to act as live human sensors. Anyone can post and report details of events they witness or experience outside in the physical world [3]. In 2020 it was discovered that almost 500 million Tweets are posted daily. This extensive daily use, speed and coverage of Twitter makes it a major social media platform and constantly a major source of data from which topical information on various events can be extracted. These events are represented by three main dimensions:

1. Time
2. Location
3. Entity-related information about the event and its participants.

We can extract all this information from the Twitter data and use it to our advantage. Figure 1 shows a sample of the information Twitter data contains and how Tweeters can act as surrogates or human sensors.

In our previous papers RISECURE: Metro Incidents And Threat Detection Using Social Media [1], we presented a tool that leverages Social Media Data and uses Tweets as surrogates to extract information relevant to any possible security events/incidents within a Metro system. Since our project started to move towards a collective disruption identification system we needed to improve our event extraction technique by using a more sophisticated model. In this paper we develop a Graph Neural Network based approach for text classification and event extraction. The graph neural network learns features by capturing information from it's neighbors [4]. Our proposed model involves building a single diverse text graph for the whole training corpus. The graph is developed by using the corpus from a pre-existing disruptions data set. This graph contains both word nodes and Tweet nodes allowing us to explicitly model the global word co-occurrence. After the graph is built it is fed into a convolutional neural network architecture which follows an approach similar to the work of Kipf et al.[4]. This architecture allows the

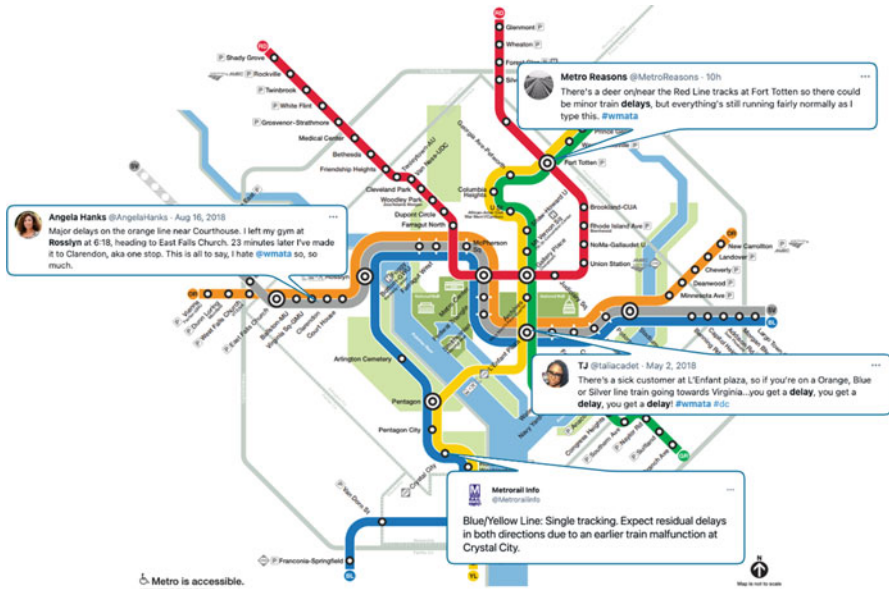


Fig. 1 An example of how Tweeters can be used as human sensors and the disruption related information contained within Twitter data

model to scale linearly in the number of graph edges by learning the hidden layer representations that encode both local graph structure and features of nodes [4].

We will provide an overview of both Dynamic Query expansion and GCN approaches in this paper. We use the same Geo-tagging technique as in our previous paper [1]. Before we integrate a new model into our system we compare the approach with other existing text classification techniques to perform a benchmark evaluation. The major contributions of this paper are:

- **Social Media Mining:** Acquisition of Twitter data to store data on the events and use it to extract candidate Tweets using keywords detection, and using Dynamic Query Expansion to track any new and emerging chatter on the incident via Dynamic Query Expansion.
- **Graph Convolution:** We Develop and implement a GCN model to classify disruptions from Twitter data and their comparative analysis. This involves building a text graph to learn feature information from the available corpus. The proposed approach is discussed after the overview of our current system, and compared to other forms of text classification.
- **Web and Mobile Platform Generation:** A convenient provision of the Data Mining model providing users with an effective visualization of the location of the event along with any necessary information in the form of a timeline.

2 Related Work

There has been a lot of work done in the field of text classification and event extraction from social media data.

Similar to our work, Ji et al. [5] approach the disruption detection problem in transit service using Twitter data. However, they utilized a multi-task learning framework in their approach. They developed a supervised model which utilizes unique metro specific assumptions in a feature space, reflected in the two kinds of regularizers proposed in the model. They proposed an algorithm based on the ADMM framework which divides the problem into a set of sub-problems which are solved using block coordinate descent and proximal operators.

Gu et al. [6] developed a technique to mine Tweets to extract traffic incident information on highways and arterials. They developed a dictionary of important keywords and used combinations of those keywords to detect traffic incident information. Tweets were mapped to a binary vector in a feature space formed by the dictionary and labeled as incident related or not. If they were labeled as traffic incident related, they were geo-coded and further classified into the respective incident classes. Zhang et al. [7] assessed the use of Tweets for traffic incident awareness. They developed a Latent Dirichlet Allocation (LDA) model and document clustering technique to model incident-level semantic information and also applied spatial point analysis to explore certain spatial patterns.

Traditional forms of text classification use various feature engineering techniques. Several techniques build text representations after learning word embeddings [8–10]. With development of neural networks people have used Convolution Neural Networks for sentence classification [11] and Recurrent Neural Networks for text classification using multi task learning frameworks [12]. There have been several studies where various researchers tried to develop a more general architecture similar to a CNN model that could work on arbitrary graphs. One such implementation is presented by Kipf et al. [4], who were able to use a GCN to outperform other techniques in several tasks including text classification, machine translation etc.

3 System Overview

In this section, we illustrate the system architecture of the RISECURE application, as pictured in Fig. 2. The GCN model integrated application follows a similar architecture, where instead of the query expansion module we integrate our GCN.

3.2 Application Server

This is the core server component of the application. We use AWS and MongoDB to help integrate the data acquisition module and the backend database. After the data has been acquired, the AWS Lambda function will trigger and send an API request to our backend service to update the data in our database. For the backend, we use Express.js with node.js as a web server framework following REST API principles.

3.2.1 Application and Mobile Interface

This is the major component of user interactions and operations. The web application was built based on the React.js framework and Google Map API. Besides, we use Progressive web application(PWA) to construct our mobile app. PWA can be installed on the user's device much like native apps and provide cross-platform compatibility for iOS and Android. The disruption incidents are accessible from the UI through 3 major components: the real time panel, the station marker and the alert notification pop ups.

3.2.2 Real Time Incidents Panel

The real time panel provides the user with the latest information about any occurring incidents at any station. Tweets related to incidents are collected by timestamp and are used to construct a real-time storyline. Each incident related Tweet is tagged under a specific category which is displayed on the yellow label. The user is also provided a link to the original Tweet itself. Figure 5 provides a concept of the real-time panel.

3.2.3 Alert Notification System

The alert notification system allows users to subscribe to multiple stations and the system provides an immediate alert notification when an incident is detected. The alert notification system also updates the latest follow-up information once the authority validates the authenticity of the event. Figure 6 illustrates the scenario of our application pushing an alert notification for first event-related Tweets posted and then the verified event notification.

3.2.4 Station Marker

Station markers with a red warning sign indicate a disruption at the station due to a security incident. Clicking on the station marker will display two small pop ups.

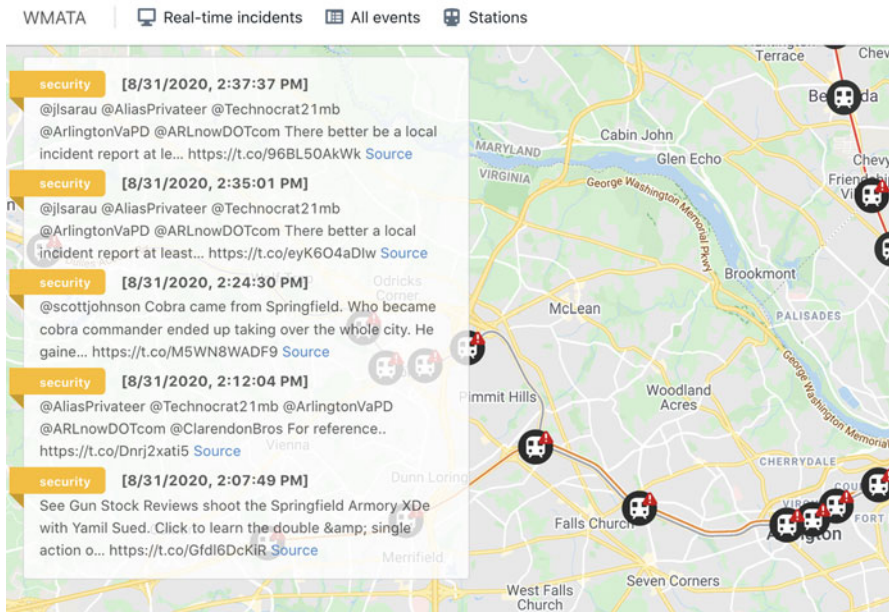


Fig. 5 Real Time Incidents Panel

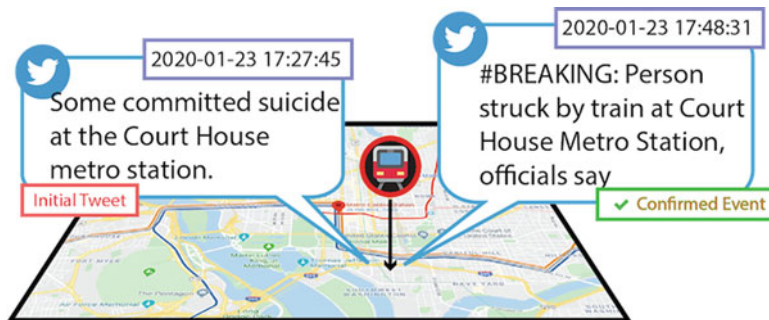


Fig. 6 Alert notification system

One shows the details of the station itself and the other displays a timestamped storyline which contains events specific to that particular station. This allows users to navigate to the station of their choice on the map and stay updated with any recent incidents at that station. Figure 7 shows a concept of this component.

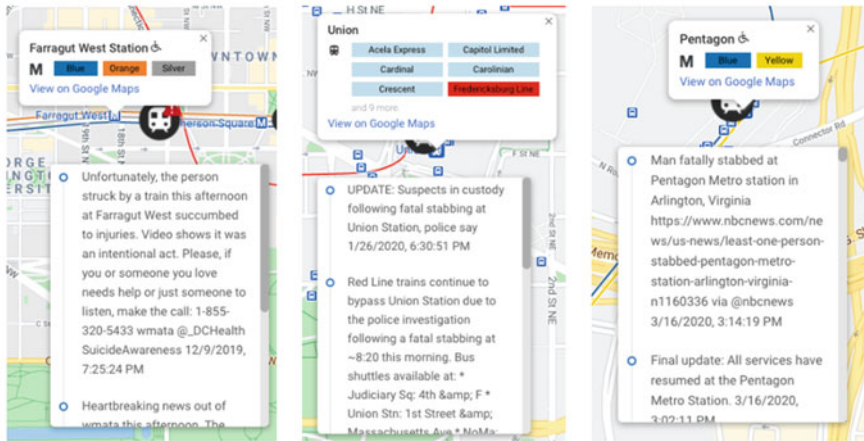


Fig. 7 Station event list

4 Methodology

This section will go over the details of the two proposed models for this study.

4.1 Dynamic Query Expansion

After the Twitter stream returns a pool of WMATA related Tweets, a second query of keywords is used to filter out Tweets that maybe related to disruption. This query consists of keywords that were found to be able to identify incidents that may cause disruptions in the service or jeopardize the safety of the commuters and the infrastructure of the system. Table 1 shows the keywords used to identify these Tweets. This query then returns Tweets similar to the ones shown in Fig. 1. Once a disruption related Tweet is found, the dynamic query algorithm is run on that Tweet to track incident and retrieve updates.

Table 1 Keywords used for disruption query

| |
|---------------------|
| Disruption keywords |
| Police |
| Malfunction |
| Slowdown |
| Delay |
| Brake |
| Fire |
| Emergency |
| Bypass |
| Single track |
| Uncoupled |
| Injury |
| Crash |
| Struck |
| Investigation |
| Disabled |
| Power outage |
| Operational |
| Door |
| Signal |
| Rush hour |

Dynamic Query Expansion evaluates inputs and reformulates the query result to improve retrieval performance. After acquiring the candidate Tweets, we can use data to extract the representative keywords for a specific threat event. This helps keep track of the emerging information for the event as it progresses. Besides, we select some high-frequency keywords, as shown in Table 1, as our initial seed query S based on analyzing the historical data of threat-related Tweets.

Algorithm 1: Dynamic Query Expansion Algorithm

Input: A time-ordered sequence of Tweets $\langle T_0, T_1, \dots, T_t \rangle$, Seed Query S

Output: Expanded Query Q

Set $Q_0 = S = F_0, w(F_0) = 1, k = 0$

repeat

$k = k + 1;$

$w(F_k) = idf(F_k) \cdot C \cdot w(T_{(k-1)})$

$w(T_k) = \Phi \cdot C' \cdot w(F_k);$

repeat

$swap(\min(w(T_k)), MAX(w(T - T_k)));$

$\sigma = \min(w(T_k)) - MAX(w(T - T_k));$

until $\sigma \leq 0;$

until $w(F_k) = w(F_{(k-1)});$

$Q = F_k;$

To select the representative keywords, we use the algorithm based on Dynamic Query Expansion (DQE) techniques [14, 15]. Given a time-ordered sequence of Tweets $\langle T_0, T_1, \dots, T_t \rangle$ and Seed Query S , we could retrieve the new expanded query Q to represent this event. F_k is the feature node. W is the set of weights for nodes where higher weights denote a higher degree of relation between the node (either a Tweet or a feature) and threat-related theme. We can calculate the weight of F_k by Inverse Document Frequency (IDF) and weight of $T_{(k-1)}$. C is the adjacency matrix.

For each iteration, dynamic query expansion compares the minimum weight of the related Tweet node and the maximum weight of unrelated Tweet node, selecting the one with a higher score and putting it in the result. After the k th iteration, it converges to the stable representative keywords. After the stable status is reached, we can assume that the highest weighted keywords could describe the event. We retrieve this result and represent it on our application. The dynamic query expansion for the Pentagon Metro Stabbing Case study is shown in Fig. 8. After more event-related Tweets are collected, we can see how keywords transform from an initial query with equal weight to the expanded query with more representative keywords. The algorithm detected an initial tweet of an African American male being stabbed at Pentagon Station around 9 AM using the initial query. Over the course of the next few hours as new information comes in, the query expansion is at work. We see the word Pentagon add to the expanded query after the initial tweet, helping us identify the location of the incident. As more and more data comes in we also collect information about the disruptions caused by the incident. We see the algorithm collect information about delays on the blue and yellow lines due to an ongoing police investigation.

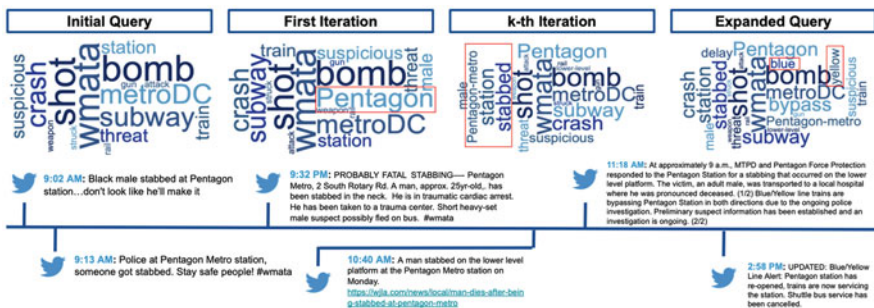


Fig. 8 Dynamic query expansion for pentagon metro stabbing case study

4.2 Graph Convolution

Graph Neural Networks have been commonly used for classification techniques lately. A GNN is a model that is built on the concepts of Graph Theory. A graph is a type of data structure that allows one to easily represent the relationships between different types of data it contains through nodes and edges. In the graph, a data point is represented as a node and edges connect or link multiple data points. The weights of the edges and distances between nodes define the relationship amongst the data. This text classification experiment that we are performing turns into a node classification for our proposed approach.

In text classification the nodes represent individual words or documents and the edges represent the word co-occurrence in the document or corpus depending on the type of edge. The graph is translated into a feature network and is represented as an Adjacency Matrix. For a Graph Convolutional Network the convolution operation is similar to that of a regular Convolutional Neural Network where the model learns the features by inspecting neighboring nodes in the graph. The GCN will take a weighted average of neighbor node features, including itself. The resulting feature vector is then passed through a neural network for training which learns the relationship amongst the data for classification. This neural network then returns a final vector containing the result of the classification. Rather than just identifying keywords like the dynamic query expansion technique, the GCN has a contextual learning ability which gives us the advantage against other techniques.

The GCN model here in (Fig. 9) is responsible for gleaning insight from the collected disruptions data and identifying disruptions in a live Twitter stream. It is setup, trained and developed in the following manner:

1. The training data is first cleaned and pre-processed to remove any unnecessary non alphabetic characters.
2. The word embedding or graph is generated from the training corpus. The graph can contain word nodes and Tweet nodes.



Fig. 9 GCN setup

3. This graph is then passed to a neural network which is trained by learning the relationships in the graph and then we test our model.
4. The final model is then tested by comparing it to other pre-existing Graph based text classifiers and commonly used text classification techniques. It is also tested for affect on the accuracy by tuning the parameters.

4.2.1 Building The Graph

The main component of this Tweet-GCN is the word embedding developed to train the model. A large and diversified text graph is built which contains word nodes and Tweet nodes. This allows us to explicitly model the global word co-occurrence so that graph convolution can be easily performed. In this graph, the number of nodes $|V|$ is the total number of Tweets plus the total number of unique words in the training corpus. This is just the total corpus size combined with the total vocabulary size after the preprocessing stage. A feature matrix M is set as an identity matrix ($M = I$). This means that every word and Tweet is represented as a one hot vector as the input for the Tweet-GCN. A one hot vector is just representation of categorized variables in the form of binary vectors. This helps the machine recognize categorized data much better.

The edges are built among nodes based on two properties:

1. **Tweet to Word Edges:** Based on the word occurrence or frequency in Tweets.
2. **Word to Word Edges:** Based on the word co-occurrence or frequency in the entire corpus.

The weight of an edge between a Tweet node T and word node W is the term frequency inverse Tweet frequency, which is just the TF-IDF of the word W in the Tweet T . Point-wise mutual information (PMI) was used to calculate the weights for word to word nodes. We want to quantify the likelihood of the co-occurrence of two words and PMI helps with that. PMI is a popular measure used for word associations to calculate the weight between two word nodes. Using PMI instead of only using word co-occurrence helped achieve better results in the experimentation and testing stages of the model. The weight of the edge E_{mn} between node m and node n is defined as follows:

$$E_{mn} = \begin{cases} \text{PMI}(m, n) & m, n \text{ are words, } \text{PMI}(m, n) > 0 \\ \text{TF-IDF}_{mn} & m \text{ is Tweet, } n \text{ is a word} \\ 1 & m = n \\ 0 & \text{otherwise} \end{cases}$$

The PMI value for a word to word edge E_{mn} is calculated using the following equations:

$$PMI(m, n) = \log \left(\frac{p(m, n)}{p(m) * p(n)} \right) \quad (1)$$

$$p(m, n) = \frac{\#win(m, n)}{\#win} \quad (2)$$

$$p(m) = \frac{\#win(m)}{\#win} \quad p(n) = \frac{\#win(n)}{\#win} \quad (3)$$

where $\#win(m)$ or $\#win(n)$ represents the number of sliding windows in the corpus that contain the words m and n respectively. $\#win(m, n)$ is the number of sliding windows in the corpus that contains both m and n , and $\#win$ represents the total number of sliding windows that are present in the corpus. The results from the PMI equation tell us about the semantic correlation of the two words in the corpus. A positive value indicates a high correlation, a negative value indicates little or no correlation and a value of 0 indicates that the two are statistically independent. Due to this nature, only edges with a positive PMI value were added to the graph.

4.2.2 Network Architecture

Once the graph has been built, it is fed into a multi-layer neural network architecture which follows a similar approach to Kipf et al. [4]. This architecture performs convolutions directly on the graph by inducing embedding vectors of nodes based on the properties of their neighboring nodes. The graph G can be represented using adjacency matrix A and degree matrix D . Using a single layer of convolution will allow the GCN to only capture the information from its immediate neighbors. Stacking up multiple layers gives the GCN the ability to obtain information over larger neighborhoods in the graph. For a single layer architecture, the new d dimensional node feature matrix X is computed as:

$$X_d^{(1)} = \phi(\hat{A} * X * W_0) \quad (4)$$

where \hat{A} is a re-normalized adjacency matrix and W_0 is the weight matrix. ϕ is a rectifier activation function (ReLU) where $\phi(x) = \max(0, x)$. For a multi-layered architecture, X is computed as:

$$X_d^{(i)} = \phi(\hat{A} * X^{(i)} * W_i) \quad (5)$$

where $X^{(0)} = X$ and i denotes the number of the current layer. For this model a two layer architecture was used. The word and Tweet node embeddings in the second

layer have the same size as the labeled data set. These embeddings are then passed to a softmax classifier function:

$$y_{pred} = softmax(\hat{A} * ReLU(\hat{A}XW^{(0)})W^{(1)}) \tag{6}$$

where W_0 is an input to hidden layer weight matrix for any hidden layer with H feature maps and $W^{(0)}$ is the weight matrix for the hidden to output layer. The loss function is defined as the multi class cross entropy over the entire labeled data set.

$$L_{crossEntropy} = - \sum_{i=1}^{C_o} v_0 * \log(y_{pred}) \tag{7}$$

C_o represents the number of possible output classes, v_0 denotes one-hot encoded representation of the ground truth label and y_{pred} is the is the probability of the predicted label for the Tweet. The weight parameters were $W^{(0)}$ and $W^{(1)}$ were trained by performing batch gradient descent using the full data set for each iteration. The only downside to this approach is that it requires a large amount of memory to train the model. The two layer GCN allows for message passing between nodes that are two edges or steps apart from each other. So even though there are no predefined Tweet to Tweet edges in the graph, the two layer GCN allows pairs of Tweets to exchange information between each other.

Figure 10 shows an overview of the Tweet-GCN model. Nodes beginning with T are Tweet nodes and the rest are word nodes. Tweet to word edges are denoted by the solid black lines and word to word edges are denoted by solid red lines. $E(x)$ here denotes the embedding representation for x . For example, $E(track)$ is the embedding representation of the word track. The different colors here indicate the different classes of disruptions in the data set. Only four classes have been shown here to avoid a cluttered schematic.

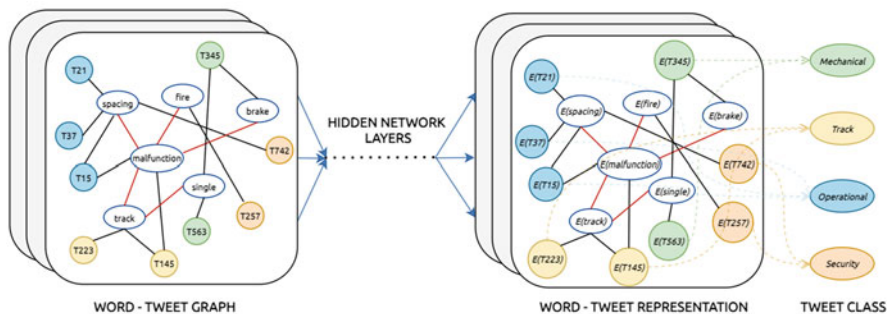


Fig. 10 Overview of Tweet-GCN

5 Experiment and Results

Our data set included 60000 disruption incidents from between 2012–2019. We also collected data from WMATA service reports which was used as the ground truth for labeling our Twitter data.

5.1 Benchmark Evaluations

We fine tuned our parameters accordingly to achieve the best results in each case, which are discussed later. For our GCN we started by setting the learning rate as 0.02, dropout rate to 0.5, the embedding size of the convolution layer to 200 and the window size to 15. 10% of our training data was randomly selected as our validation set. We used the following baseline models:

1. **TFIDF + Regression:** Look at a basic bag of words(BOW) model with a Logistic Regression classifier.
2. **LSTM:** The LSTM model implemented by Liu et al. [12]. It uses the previous hidden state as the representation of the entire text, with and without pre-trained word embeddings.
3. **CNN:** Using the Kim et al. [11] implementation of the Convolutional Neural Network which uses max pooling on the embeddings to generate the text representations. The non-static CNN approach is used, which uses pre-trained word embeddings.
4. **Graph CNN-C:** A graph CNN model by Defferrard et al. [16] that operates convolutions over word embedding similarity graphs by utilizing a Chebyshev filter.
5. **Graph CNN-F:** A graph CNN model by Henaff et al. [17] which is similar to Graph-CNN C but instead utilizes a Fourier filter.

For the Tweet-GCN we used the optimal parameters based on the results of our parameter tests and for the baselines the default parameters were used as discussed in the papers of their original implementations along with the pre-trained 300 dimensional Stanford NLP GloVe word embedding where ever necessary. 10% of the training set was randomly selected as the validation set and following the approach of Kipf et al. [4] both GCNs were trained for a maximum of 200 epochs with early stopping enabled for no changes in validation loss for 10 consecutive epochs.

Tweet-GCN Setup A dimension size of 200 was set, a window size of 12, a learning rate of 0.02, dropout probability of 0.5 and an L2 regularization or weight decay of 0.

Four different metrics were used to evaluate the performance of the proposed approaches against the baselines. These were accuracy, precision, recall and F-score. The weighted average of all four metrics across 100 Runs for each model was

reported. The accuracy measure is one of the most common metrics for evaluation. Generally, higher the accuracy is, the better the classifier is at identifying class labels. Precision gives us a measure of the relevant data points by identifying what proportion of the predicted positives is truly positive. Recall gives us a measure of how accurately the model is able to identify the relevant data points by telling us what proportion of true positives have been correctly classified. Most of the times there is a trade-off between the precision and recall scores. Sometimes these values may conflict, so they must be considered comprehensively. For this problem we would prefer to have a good value for both measures since we want to be able to identify as many disruptions as possible as precisely as possible. That is where the F-score comes in, which is the harmonic mean of the precision and recall. The F-score also does a better job at describing models dealing with class imbalance in multi-class classification problems.

From Table 2 it can be seen that all the graph based models outperform the rest of the models. This is likely due to the characteristics of the graph structure enabling the word nodes to learn the representations more accurately. Something which is impossible for the other traditional models. We see lower accuracy results and testing scores for the Dynamic Query Expansion model compared to the baselines. The Dynamic Query Expansion and keyword extraction model does not have a learning ability like the graph based models and other deep learning models therefore it is at a disadvantage when it comes to learning the semantic relationship among the data and capturing relevant disruptions data within the stream. The Tweet-GCN performs well because the graph is able to capture both Tweet-word relations and global word-word relations and because the label information of the Tweet nodes can be passed to the adjacent word nodes and relayed to other word and document nodes that are at most two steps away. This allows the Tweet label information to be propagated throughout the graph.

Graph CNN-C and Graph CNN-F use similar graph models as ours but the word nodes are connected over larger windows without weighted edges. Due to the lack of trainable edges those models are then unable to learn the significant relationships between different words. We also notice that the CNN and LSTM models provide

Table 2 Test scores for disruption classification with for the twitter disruptions data set: the results are the average of 100 runs for each model

| Model | Accuracy | Precision, Recall, F-Score |
|------------------------------|---------------------|----------------------------|
| Logistic Regression + TF-IDF | 0.8101 \pm 0.0018 | 0.81, 0.80, 0.81 |
| LSTM | 0.7743 \pm 0.0087 | 0.77, 0.78, 0.78 |
| LSTM(GloVe) | 0.8237 \pm 0.0163 | 0.81, 0.82, 0.82 |
| CNN(GloVe) | 0.7781 \pm 0.0048 | 0.78, 0.80, 0.79 |
| Graph CNN-C | 0.8204 \pm 0.0032 | 0.82, 0.78, 0.80 |
| Graph CNN-F | 0.8371 \pm 0.0015 | 0.83, 0.84, 0.83 |
| Dynamic Query Expansion | 0.7892 \pm 0.0153 | 0.75, 0.79, 0.77 |
| Tweet-GCN | 0.8726 \pm 0.0021 | 0.87, 0.86, 0.87 |

satisfactory results on both datasets, but lack in their contextual information learning ability compared to the our GCN. However, both those models use pre-trained word embeddings while our GCN only uses the information provided to it by the input corpus.

5.2 Parameter Testing

We also performed tests by varying the parameters of our GCN.

5.2.1 Size of Sliding Window

In Fig. 11 we have the results from varying the sizes of sliding windows in the model.

We see the test accuracy be the highest for a window size of 15, and the accuracy begins to decrease when the window size becomes larger than that. Small window sizes are unable to generate enough word co-occurrence while information too large window sizes are may add extra edges to nodes that might not be closely related [4]. Since Tweets are a form of micro-blogs or short text this behavior is understandable, suggesting that a small window size may not be able to capture enough information

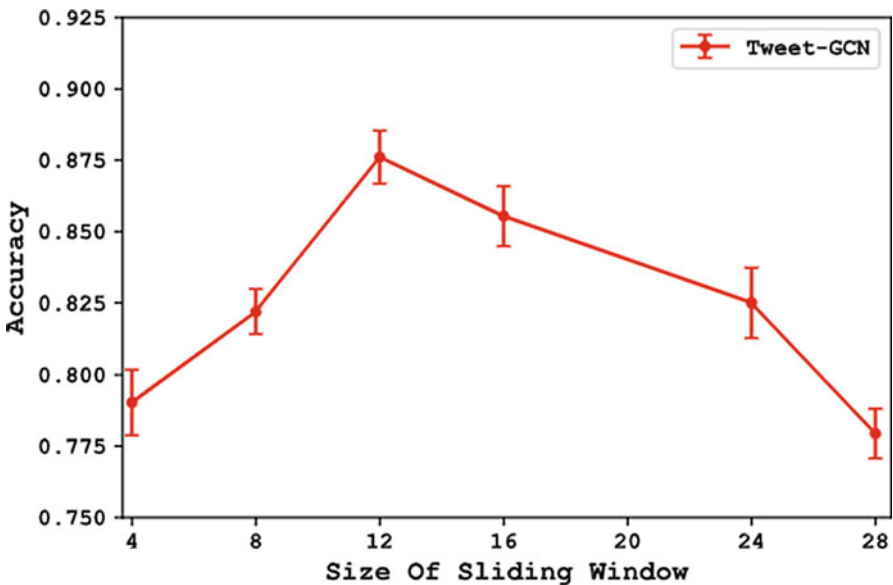


Fig. 11 Accuracy with varying window sizes

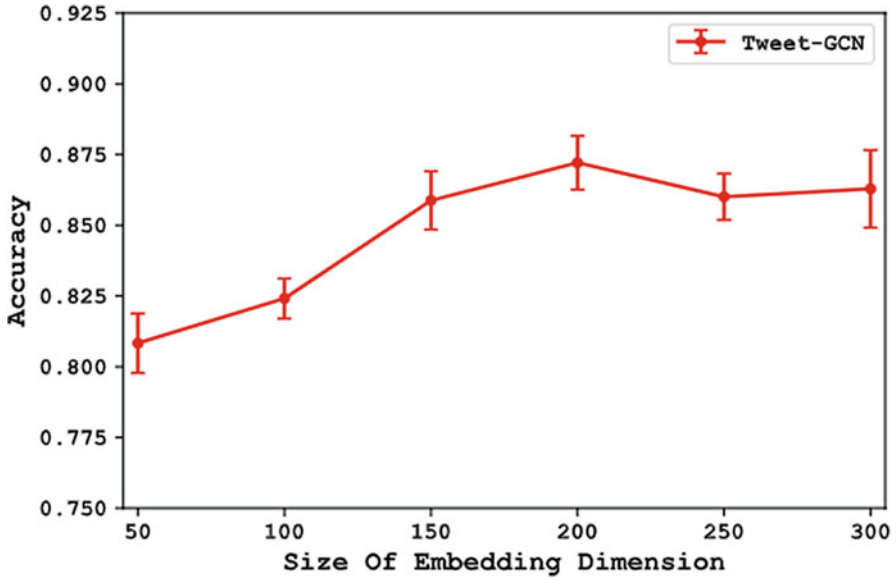


Fig. 12 Accuracy with varying embedding sizes

where as a large window size may capture extra information by adding extra edges between nodes that might not be closely related.

5.2.2 Size of Embedding Dimension

Figure 12 shows the test accuracy results with varying embedding sizes. We observed trends similar to our earlier test with the sliding windows. For small dimensions, the embeddings may not disseminate throughout the graph while large embeddings increase training time and do not change our results by much. We found a dimension size of 200 to be optimal for both data sets.

5.2.3 Size of Training Data

To avoid a cluttered graph, we selected the best individual performing models to see how changing the size of our labeled training data effects the models. Figure 13 shows the test accuracy results of these tests on 1%, 5%, 10%, 15%, 20% and 25% of the Twitter training data set.

It can be seen that Tweet-GCN performs better by achieving higher test accuracies throughout for the partial training set. We can see the Tweet-GCN achieves an accuracy of 0.8132 ± 0.0132 for only 25% of the training data set. These results are even higher than those of some baseline models when they were trained

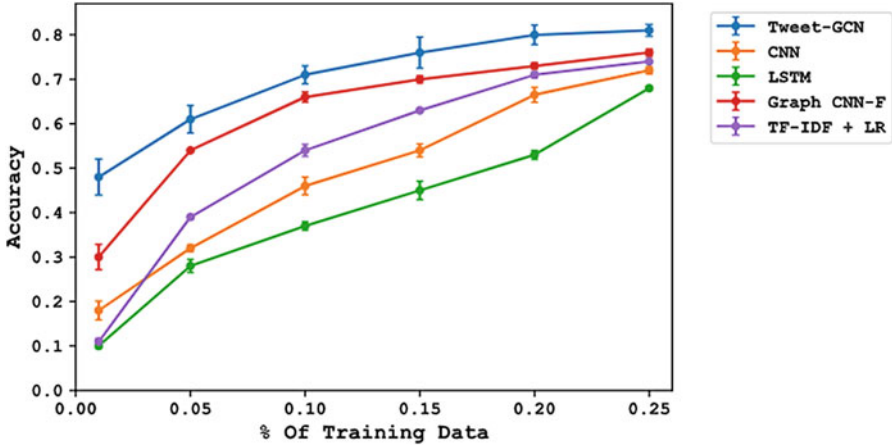


Fig. 13 Effect of size of Twitter training data on model performances

using the entire training set. These results suggest that the proposed GCN model perform reasonably well with a limited label rate and can spread and preserve label information within the graphs giving themselves the upper-hand at identifying disruption data.

From the results of our experiment we can see that our GCN model is able to achieve pretty good test results for classifying WMATA related disruptions within Twitter data. However there are still some limitations with the GCN when it comes to unlabeled information in the training data. Overcoming this hurdle will be part of some future work we look to accomplish.

6 Conclusion

RISECURE is an open-source and automated system that is capable of detecting transit disruptions by using Social Media data mining and deep learning techniques. We proposed two approaches; Dynamic Query Expansion and Tweet-GCN. The effectiveness of our proposed approaches is displayed through benchmark evaluations against other baseline models. We saw the Tweet-GCN give us the best results for identifying disruptions with an overall accuracy of 87.3%, whereas the Dynamic Query Expansion model delivered the lowest scores with an overall accuracy of 78.9%. However, the Tweet-GCN consumes a lot of memory due to the high number of edges in the corpus level graph. For future we will look to modify the model to solve this issue. For real-world deployment in transit systems such as metro rails, our proposed approach can serve as a supplementary resource to aid in swift disruption detection and, gain situational awareness. We foresee a great potential to take this

platform to a higher level where it can help improve the rider experience for the public transit systems.

References

1. Zulfiqar O, Chang Y-C, Chen P-H, Fu K, Lu C-T, Solnick D, Li Y (2020) Riscure: metro incidents and threat detection using social media. In: Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)
2. Metrorail ridership grew by 20,000 trips per weekday in 2019 (2020). <https://www.wmata.com/about/news/2019-Metrorail-ridership.cfm>
3. Zheng Y, Capra L, Wolfson O, Yang H (2014) Urban computing: concepts, methodologies, and applications. *ACM Trans Intell Syst Technol* 5:13838–13855
4. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. *CoRR* abs/1609.02907. [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
5. Ji T, Fu K, Self N, Lu C-T, Ramakrishnan N (2018) Multi-task learning for transit service disruption detection. In: Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, New York, pp. 634–641
6. Gu Y, Qian S, Chen F (2016) From twitter to detector: Real-time traffic incident detection using social media data. *Transportation Research Part C: Emerging Technologies* 67:321–342. <https://doi.org/10.1016/j.trc.2016.02.011>
7. Zhang S (2015) Using twitter to enhance traffic incident awareness. In: Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp 2941–2946. <https://doi.org/10.1109/ITSC.2015.471>
8. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International Conference on Machine Learning. PMLR, pp 1188–1196
9. Joulin A, Cissé M, Grangier D, Jégou H et al (2017) Efficient softmax approximation for gpus. In: International Conference on Machine Learning. PMLR, pp. 1302–1310
10. Tang J, Qu M, Mei Q (2015) Pte: Predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1165–1174
11. Kim Y (2014) Convolutional neural networks for sentence classification. *CoRR* abs/1408.5882. [arXiv:1408.5882](https://arxiv.org/abs/1408.5882)
12. Liu P, Qiu X, Huang X (2016) Recurrent neural network for text classification with multi-task learning. In: Twenty Fifth International Joint Conference on Artificial Intelligence
13. Gurreiro M, Megler V (2020) Detect change points in your even data stream using Amazon Kinesis Data Streams, Amazon DynamoDB and AWS Lambda. Amazon, Washington
14. Zhao L, Chen F, Dai J, Hua T, Lu C-T, Ramakrishnan N (2014) Unsupervised spatial event detection in targeted domains with applications to civil unrest modeling. *PloS One*, 9(10), e110206
15. Khandpur RP, Ji T, Ning Y, Zhao L, Lu C-T, Smith ER, Adams C, Ramakrishnan N (2017) Determining relative airport threats from news and social media. In: Twenty-Ninth IAAI Conference
16. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR* abs/1606.09375
17. Henaff M, Bruna J, LeCun Y (2015) Deep convolutional networks on graph-structured data. *CoRR* abs/1506.05163

Local Taxonomy Construction: An Information Retrieval Approach Using Representation Learning



Mayank Kejriwal, Ravi Kiran Selvam, Chien-Chun Ni, and Nicolas Torzec

Abstract In specific domains, such as e-commerce, inducing a taxonomy over a given set of ‘target’ concepts is an important problem with applications ranging from good website design to knowledge organization and recommender systems. Automatically inducing a full or ‘global’ taxonomy over a large set of concepts is a difficult problem in the AI literature, typically requiring human intervention. A more tractable version of the problem is called Local Taxonomy Construction (LTC). Rather than induce a global taxonomy, LTC attempts to induce the local neighborhood of each concept in the target concept-set. Despite having much practical importance, LTC has not been properly formalized and explored in the applied AI community. In this paper, we present such a formalism on LTC, including a set of viable, minimally supervised solutions based on pre-existing representation learning algorithms in the natural language processing community. We also conduct a detailed experimental study using three widely used, real-world e-commerce datasets. We also present qualitative assessments, and discuss potential applications of LTC in industry.

Keywords Taxonomy induction · Local taxonomy construction · Information retrieval · Representation learning · Concept ranking · E-Commerce · Embeddings

Kejriwal and Selvam are associated with the Information Sciences Institute at the University of Southern California, where the research was conducted. Ni and Torzec were affiliated with Verizon Media when the work was performed, but are currently affiliated with Yahoo.

M. Kejriwal (✉) · R. K. Selvam · C.-C. Ni · N. Torzec
Information Sciences Institute, University of Southern California, Marina del Rey, CA, USA
e-mail: kejriwal@isi.edu; rselvam@isi.edu; chien-chun.ni@verizonmedia.com;
torzecn@verizonmedia.com<https://usc-isi-i2.github.io/kejriwal/>

1 Introduction

In many commercial domains,¹ such as e-commerce and media, with large and heterogeneous datasets, users and needs, designers of websites, user-facing systems and even backend recommender systems must start from a set of semantic categories or *concepts* that are then structured into a proper taxonomy [3, 38]. A simple example of *taxonomy induction* or construction is illustrated in Fig. 1 [23, 32]. Assuming that we start with concepts such as *Dresses* and *Overalls*, a taxonomy needs to be induced over these concepts. Such a taxonomy could serve as the backbone of an ontology that is eventually used to design a rich user experience, but it is also useful for knowledge management and organization within an enterprise. While a final ontology would be expected to be more ‘graph-like’ and contain other ontological components, such as constraints, a knowledge engineer would not have to begin its construction from scratch if an initial taxonomy is available [7].

More generally, in the e-commerce domain, concepts are equivalent to *product categories*. Many price shopping and comparison websites pull in such concepts by the thousands from multiple websites. Relational ordering between these concepts is necessary, for developing a better understanding of, and organizing knowledge about, the domain. However, in real-world scenarios, manual induction of taxonomies from a ‘target’ concept-set is difficult because there may be thousands, or even tens of thousands, of concepts that need to be relationally ordered and structured. Even if we could automatically score each taxonomy on the basis of quality, the search problem is computationally hard because the total number of possible taxonomies is exponential in the size of the concept-set.

A related problem to taxonomy induction is *link prediction*, which frequently emerges in the case of graph-theoretic domains like social networks and ‘knowledge graphs’ [12–14, 35, 36]. However, taxonomy induction is a more difficult problem because it falls under a class of machine learning problems that have to work without any examples. Other examples of such problems include community detection [18], clustering-based applications and zero-shot learning [37]. However, taxonomy induction is different from these problems because highly localized links need to be discovered for each concept in the target set. An alternate way to differentiate between taxonomy induction and clustering is that, for the latter, the number of clusters is typically a small constant,² rarely exceeding 100, while for the former,

¹ This paper is an extended version of [15], which was an 8-page paper published in the proceedings of (and presented in the industrial track of) the 2020 IEEE/ACM ASONAM conference (held virtually). In this article, we significantly expand upon the theory and formalism of the local taxonomy construction (LTC) problem, particularly from an information retrieval (IR) approach. We also present more experimental results, visualization and analysis. The paper also contains 3 new sections not included in [15], including a *Discussion* section.

² Furthermore, this number grows very slowly, if at all, with the number of data points. Even with millions of data points, not many clusters are required to uncover structure, when using algorithms like k-Means.

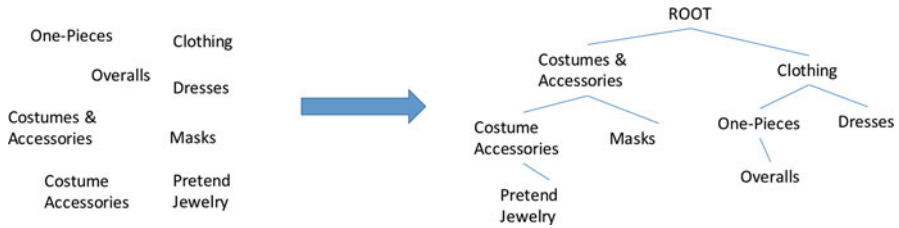


Fig. 1 An illustration of the taxonomy induction problem, using real data from the Google Product Taxonomy (Sect. 5)

the number of edges that need to be inferred linearly depends on the number of concepts in the target set.

Because automatic ‘global’ taxonomy induction, where the input is a set of concepts, and the output is a complete taxonomy, is a difficult problem, especially in domain-specific settings, this chapter addresses a more tractable version of the problem called *Local Taxonomy Construction* (LTC). Intuitively, LTC is the problem of *independently* inferring the neighbors of every concept $c \in C$, where C is a given ‘concept-set’ (set of *target* concepts). In addition to C , a practical versions of the problem may also rely on generic ‘background’ resources available on the Web, such as a domain-specific text corpus or an example taxonomy from a different website or data source. The independence assumption is important as it gives the problem its local flavor. The problem is formalized further in Sect. 3. Specific contributions of this chapter are enumerated below:

- We define and formalize the Local Taxonomy Construction (LTC) problem by using a methodology grounded in the Information Retrieval (IR) community. We also explore IR-based metrics for evaluating potential solutions to the problem.
- We explore the possibility that word representation learning or ‘word embedding’ models can be used in a variety of potential taxonomy induction scenarios, depending on the availability of background resources. We present several such solutions. Our goal is not to present a radically new algorithm but to explore how established representation learning algorithms, including a *retrofitting* algorithm can be adapted to yield good results on this problem [6, 25].
- We conduct a principled and detailed empirical study, using three real-world e-commerce concept-sets and taxonomies, and the presented representation learning solutions. Based on 20 GB+ worth of experimental data collected and analyzed, we show that such solutions offers significant promise for LTC, and can be feasibly applied on new concept-sets, at least within the e-commerce domain.
- Based on our findings and industry knowledge, we describe a list of potential applications for which LTC is particularly well-suited. We also describe promising avenues for future research, a key element of which is to design a similar formalism and experimental study for the (more difficult) global taxonomy construction problem.

The rest of the chapter is structured as follows. Section 2 provides an overview of related work, including work in representation learning, structured data embeddings, natural language processing, and other allied sub-areas that are relevant to the lines of inquiry in this paper. Section 3 provides an Information Retrieval-based formalism of the LTC problem. Section 4 describes minimally supervised solutions to the problem using established algorithms and background resources from the ‘word embedding’ literature in natural language processing. Section 5 describes the empirical study, including methodology and metrics, the three real-world e-commerce datasets we used in the study, and a detailed set of results and interpretation. Section 6 places the results in broader context by supplementing them with additional qualitative assessments. Section 7 describes potential applications of LTC, especially in the context of the minimally supervised representation learning-based solutions that we presented. Section 8 enumerates avenues for future research. Finally, Sect. 9 concludes the paper.

2 Related Work

The research presented in this paper is related to several existing lines of research that we briefly cover below.

Representation Learning and NLP With the advent of neural networks and deep learning in the last decade, representation learning (also known as ‘embeddings’) have become very prominent in the Natural Language Processing (NLP) community [11, 24, 26]. The original word2vec algorithm [24] has now been in popular use for almost a decade, and relies on a relatively ‘shallow’ neural network compared to Convolutional Neural Networks (CNNs) and other such models. It learns representations of words by sliding a window of fixed size (say, 5 words) over the words in the corpus, represented as a sequence, and optimizing an objective function whereby words that tend to fall in the same window also tend to be closer together in the vector space. GloVE follows a similar philosophy, but is based on constructing a word co-occurrence matrix and relies on a slightly different objective function and formalism [26]. The FastText algorithm (also called ‘bag-of-tricks’) [11], developed by the same researchers in [24], was publicly released by Facebook AI research and improves on previous models by being able to robustly handle misspellings and other variations in words. Consequently, it is able to retrieve embeddings for words that it did not see during the training phase.

More recently, transformer-based models such as BERT have become popular [4], although much more expensive to train on a corpus (and requiring far more data) than previous algorithms such as fastText [11] and Glove [26]. Intriguingly, the ability of the previous word embedding algorithms to do analogical reasoning without any apparent supervision (e.g., the famous ‘King is to Queen as Man is to Woman’ example), and the ability of recent transformer-based models to achieve near-human performance on complex tasks like common sense question answering

and other NLP applications, raises the question of whether such models can be applied to solve ever more sophisticated problems. We make extensive use of the classic word embedding algorithms in this paper, while transformer-based models are left for future research, as it is not clear how one can apply them for the problem of local (or global) taxonomy induction.

Embeddings for Structured Data The success of word embeddings, and similar representation learning models (e.g., based on autoencoders and decoders [39]) in the computer vision community has led to similarly successful proposals being floated both in the knowledge discovery and Semantic Web communities. In the former, network embedding algorithms such as DeepWalk have become popular and are based loosely on word2vec [27]. Given a network, modeled as a graph, DeepWalk learns embeddings for the nodes in the graph by first executing random walks (starting from each node), and treating each random walk as a sequence that can be input into word2vec. By treating nodes as ‘words’ in this way, an embedding is learned for each node and found to perform quite well compared to other non-neural approaches.

In the Semantic Web, RDF2Vec has become a popular choice and shown to be useful in several Semantic Web problem areas [29]. Its philosophy is similar to DeepWalk, but it is more apt for ‘knowledge graphs’ (directed, labeled multi-relational graphs) represented in the Resource Description Framework (RDF) language. More generally, ‘knowledge graph embeddings’ such as TransE and HolE have become popular choices for embedding multi-relational graphs such as DBpedia and Freebase [36]. These algorithms embed both nodes and relations, and use a different optimization function compared to network embeddings such as DeepWalk. However, such algorithms invariably assume that (i) the node-set (what we call the concept-set in this paper) of the graph is typically known in advance; (ii) there exists a set of ‘positive’ links between many of these nodes, which can be used to train a model and ‘complete’ the graph by inferring more links. The problem in this paper is very different: we only consider one relation,³ rather than multiple relations, and the problem is a ‘from-scratch’ problem, since no example edges are available during test time for an algorithm to exploit.

Retrofitting Pre-Trained Embeddings While more recent embedding algorithms such as BERT rely on the notion of ‘fine-tuning’ to derive a ‘refined’ set of task-specific embeddings from a set of pre-trained embeddings⁴ [4], the problem of refinement and computational costs of re-training were realized earlier in the decade in the NLP community. Some authors showed, for example, that performance

³ One could theoretically consider the induction of a multi-relational taxonomy from scratch; we leave it as a promising avenue for future research.

⁴ Trained on a large, generic corpus of text (usually including at the very minimum, Wikipedia and/or a news corpus) that took many hundreds of hours of computational time to train even in the industrial labs where these algorithms were developed.

on various downstream tasks tended to improve when pre-trained embeddings were ‘retrofitted’ to an existing *semantic lexicon* such as WordNet [6, 25]. In this paper, we show that retrofitting can be a valuable baseline for learning domain-specific models for local taxonomy construction. To the best of our knowledge, such methods have mostly been applied to improving the embeddings for an NLP problem, and not on a taxonomy construction problem as is this paper’s subject. Retrofitting is detailed further in a subsequent section of the paper (Sect. 4).

Taxonomy and Ontology Induction Taxonomy (and more generally, ontology) induction has been studied for some time now, but the underlying data on which the taxonomy is induced tends to be textual or context-rich (and in most cases, involves a body of generic nouns, such as WordNet extensions [32]). Examples include the work by Snow et al. [32], Velardi et al. [34], Mao et al. [23], and Gupta et al. [10]. While promising, these works rely significantly on help from extra resources *during test time* for semantic understanding. The primary reason for this is that many of them preceded the advances in representation learning that occurred in the last decade. The work by Snow et al. [32] starts from the WordNet 2.1 base, which means some links are already available. In [34], the authors present *OntoLearn Reloaded*, which relies on a specific set of text documents and Web documents for context, and is not constrained by a set of specific concepts over which a taxonomy must be induced. More recently, [23] and [10] used hypernym subsequences and reinforcement learning respectively. The former considers a problem definition that is most related to this work, but the domains used in the evaluation are still very closely related to generic domains (e.g., food) rather than complex, specific domains such as e-commerce that we consider in the evaluations in this work. For near-generic domains such as food, a resource like WordNet already contains many links, which makes the problem less challenging than a typical e-commerce application.

In contrast, the problem considered in this paper starts from the concepts, which the domain expert has decided must be included in the taxonomy. A text corpus is not guaranteed, though we do consider the possibility in one set of experiments in this paper. We show, in fact, that the best approaches do not need a domain-specific corpus for training, if a sufficiently robust embedding has been trained on a generic corpus.

To the best of our knowledge, there is no current work that has proposed to use representation learning for addressing the LTC problem. For other problem domains such as information extraction and entity resolution [5, 21], similar work has shown that reasonable representation learning models can often outperform the state-of-the-art. Put briefly, information extraction is the problem of extracting named entities (such as names of people and locations), relations and even higher-order entities such as ‘events’ from a corpus of text. Most information extraction algorithms need a set of ‘concepts’ identifying the types of named entities and relations to extract, but generally, the concepts are not structured in a taxonomy. Entity resolution is the problem of determining when two strings or ‘mentions’ (such as extracted named entities) refer to the same underlying entity. For example, ‘LA’ and ‘Los Angeles’ refer to the same underlying entity.

In particular, representation learning has achieved impressive performance on problems (e.g., relation extraction) that had proven too difficult to tackle before. This was found to be especially the case in minimally supervised settings, such as involving weak or distant supervision [21]. We attempt to prove the same for local taxonomy construction through a carefully designed and executed empirical study. We also provide qualitative assessments of our results.

3 Local Taxonomy Construction

The *general* problem of taxonomy induction may be stated as one of inducing a taxonomy T given a concept-set C , potentially given some background or ‘training’ resource (or a set of resources) B . T , in this context, may be thought of as a ‘tree’ over C : $T = \{(c_i, c_j) | c_i, c_j \in C \times C, c_i \neq c_j\}$. Taxonomies could be defined in more general ways, but in practice, product taxonomies often are modeled along the lines of a tree as they are meant to guide product categorization and website navigation.⁵ Formally, to ensure that a concept can have at most one parent, we assume the constraint that if $(c_i, c_j) \in T$ and $(c_k, c_j) \in T$, then it must necessarily be the case that $c_i = c_k$. We call c_i the *parent* (equivalently, super-type) of c_j . This is an abstract relation with reasonably well-defined semantics (though usually of a domain-specific nature): in the natural language community, it tends to go by the name *hypernymy* while in the Semantic Web, the `rdfs:subClassOf` predicate is the best fit. Note that, since there may be ‘upper-level’ nodes $\{c_1, \dots, c_m\}$ that do not have parents, we assume (without loss of generality), that there is a single artificial *root* node c_R that is introduced after taxonomy induction and that serves as the parent of every concept in $\{c_1, \dots, c_m\}$.

Inducing a taxonomy given just C is a difficult problem that has not been addressed much in the literature, though the previous section covered some relevant work. In practice, inducing the full taxonomy is not necessary. Instead, we tackle the more manageable problem of determining the neighbors of a concept without being given any training data (existing neighbors). The local version of the problem is expected to be somewhat immune from the problem of *error cascading* that would ensue if we were trying to recover a single global taxonomy by *combining* local ‘fragments’ in some principled way.

With the terminology above, *local taxonomy construction* (LTC) can be stated in similar terms as full taxonomy induction (henceforth called taxonomy induction). Given a concept $c_i \in C$, an LTC approach would aim to determine the neighbors (including the super-type) of c_i in the underlying *unknown* taxonomy T (though in evaluations we only use known taxonomies as ground-truths). Similar to taxonomy induction, LTC can also draw upon a background resource B .

⁵ However, in principle, future work could consider generalizing the problem to one of ‘graph induction.’

The formulation above raises two important questions. First, what is the nature of B ? We present several choices in the next section, based on representation learning, including a domain-specific text corpus, a pre-trained embedding model and a taxonomy over the same domain but on a different concept-set. Second, regardless of an actual approach used, how do we *evaluate* T ? It is reasonable to assume that, on average, a ‘good’ LTC system would aim to retrieve true neighbors (in the underlying taxonomy) while minimizing false positives and negatives. In practice, to make a solution more robust, we would frame it in terms of *Information Retrieval (IR)*, used by all major search engines to evaluate the efficacy of their ranking algorithms [22]. Next, we describe the IR-based evaluation protocol and its rationale in more detail.

3.1 Framing LTC as an Information Retrieval (IR) Problem Instance

In *ranking-based* information retrieval (IR), which is the model adopted by search engines, e-commerce providers, and many other practical applications on the Web, a user issues a *query* q to express its intent [22]. Usually, this is just a short phrase (e.g., ‘movies currently showing in Los Angeles’). Historically, search engines treated such queries as a ‘bag of words’, thereby accounting for the total count of words (even if a word was repeated) but not the order of words in the query. More recently, with the advent of powerful language models [4] and advanced search technologies such as knowledge graphs [12, 13, 31], the order of words has also become important. Search engines like Google are able to recognize that a phrase such as ‘Los Angeles’ refers to an entity (specifically, a location) rather than a simple two-word string without semantics. Arguably, because of these semantics, as well as increased usage, data collection, and parsing of search logs and click logs, search engines have become increasingly more adept at dealing with longer and more complex queries, including well-formed questions (e.g., ‘what is the latest unemployment figure in the US?’).

It is not a completely straightforward proposition to apply this model to solving the LTC problem. Two non-trivial challenges must be addressed. First, given that ranking-based IR tends to rely on a query-document structure, what is the ‘query’ and what are the ‘documents’ in LTC? Second, given that most IR approaches rely on documents with many words, and lots of context, they are able to make good use of text analysis tools like Term Frequency-Inverse Document Frequency (TF-IDF) [22] and word embeddings [11]. What approaches can be brought to bear for LTC, where labels are so short and there is no context (in contrast with a problem like link prediction)? Below, we briefly describe how we address the first challenge, while in Sect. 4, we address the second.

Since there is no explicit user or user-issued query in LTC, we let *each* concept $c \in C$ (the input concept-set) be a query, with the other concepts ($C - c$) serving

as the *universe of documents* that must be ranked in response to c as a query. For example, suppose we are given a concept-set with 1000 concepts. To frame the LTC problem over this concept-set as an IR problem, we pick a concept from this concept-set and then rank the other 999 concepts by treating the picked concept as the query. We repeat this experiment in turn.⁶ Formally, this yields $|C|$ rankings, or equivalently, $|C|$ queries with ranked responses. As noted earlier, how the ranking can be done given just simple labels, rather than long documents, will be the subject of Sect. 4. Next, we discuss how these results (the set of $|C|$ rankings) can be evaluated.

3.2 Evaluating IR Results

IR is a well-studied field with several metrics [22]; below, we consider two important ones. Both metrics rely on an assumption that is standard in ranking-based IR problems. First, they assume that, for a given query and a retrieved, ranked list of responses to the query, the responses that are *most relevant* to the query should ideally appear at a *better ranking*⁷ than the ones that are less relevant. In other words, if irrelevant documents or less relevant documents appear before more relevant documents in the ranked list, the metric should penalize the result.

An important point to note here is where the *ground-truth* relevance labels come from. Put another way, given a concept as a query, how do we decide which of the other concepts are relevant or not relevant (or more generally, assign relevance scores)? We discuss this further when we describe experimental evaluations in Sect. 5. The intuition is fairly simple. For rigorous studies, we use concept-sets that come from real data but for which a (manually induced) taxonomy is already available. Since the global taxonomy is available, we know the neighbors of every concept c . We treat these neighbors as the relevant entries, while non-neighbors are irrelevant entries. Of course, alternate formulations, depending on the task at hand, are also feasible e.g., we could consider the inverse shortest-path length between c and another concept c' as the relevance score of c' given c as a query (and vice versa). In many e-commerce applications, the taxonomy is *directed*, and it may be that the parent category should (or should not) be given a higher relevance score than child categories.

In practice, a global taxonomy would not be available a priori, and the task would be to construct such a taxonomy, starting with local construction (as described in this work). Some manual annotation effort and sampling would then be needed, just like with other industrial applications of this nature. For example, a set of concepts could

⁶ Note that the query concepts are *not* removed from the concept-set.

⁷ In the usual case where a response is either relevant or irrelevant, the relevant responses should occur first in the list, followed by the irrelevant responses. Evaluation metrics like the (subsequently described) NDCG also account for non-binary relevance scores.

be sampled and annotators might be shown the ranked lists produced by several approaches, including a baseline. The annotators might be asked either to select the best ranking (among the ranked lists they are provided), or to label individual concepts in the ranked lists as relevant or irrelevant, possibly on a sliding (e.g., Likert [1]) scale. Although beyond the scope of this work, developing an efficient annotation protocol for gathering an informative ground-truth without exhaustive sampling is an interesting research agenda in its own right, and a possible avenue for future research. In both the IR and human-computer interaction communities [30], much research continues to be published on this issue. To ensure the rigor of our reported results, in this chapter, we only use real-world datasets for which a global taxonomy is already available.

Normalized Discounted Cumulative Gain (NDCG) The Normalized Discounted Cumulative Gain (NDCG) is an important metric in the IR community. One of its advantages is that it is able to deal with *non-binary* (or ‘soft’) relevance scores. To compute the NDCG, we first have to calculate the DCG for query q , defined by the following equation:

$$DCG_q = rel_1 + \sum_{p=2}^n \frac{rel_p}{\log_2(i+1)} \quad (1)$$

Here, rel_i is the relevance of the i th item in a ranked list of size n . In our case, this is either a 1 (if the concept at that rank is a neighbor of the query concept in the ground-truth taxonomy) or a 0. We can compute the DCG of both the actual ranking and of an *ideal* ranking (where all relevant items are ranked at the top), the latter denoted as the Ideal DCG (IDCG). The NDCG is then given by:

$$NDCG_q = \frac{DCG_q}{IDCG_q} \quad (2)$$

Since the DCG is always less than the IDCG, the NDCG is between 0 and 1. For the performance over the entire concept-set we average the NDCG over all queries. In the rest of this work, the NDCG computed for an entire dataset (and a given retrieval system or approach) is assumed to be this average.

Mean Average Precision (MAP) Intuitively similar to the NDCG in its theoretical rationale, the MAP is formally defined as the mean of the *Average Precision (AP)*, computed as follows. Given a ranked list of size n in response to query q and assuming m relevant items in the ground truth for q at ranks r_1, \dots, r_m , we compute the *precision* at rank r_i as the total number of relevant items in the ranked sub-list $[1, \dots, r_i]$ divided by r_i . For example, imagine that the first relevant item occurs at rank 5. The precision at rank 5 is then $1/5 = 0.2$. Similarly, suppose the second relevant item occurs at rank 7 (hence, the items at ranks 1–4, and also rank 6 are irrelevant for that query). Then the precision at rank 7 is $2/7 = 0.286$, and so on.

The AP for a given query q is the average of the precision computed at ranks r_1, \dots, r_m . To continue the example above, suppose there were only the two relevant items (at ranks 5 and 7) for the query. Then the AP would be $Avg(0.2, 0.286) = 0.243$. The MAP is the mean of the APs computed over the *entire* query set. More details on these metrics can be found in any standard reference on IR [22]. For each of the two metrics, we obtain a single number between 0 and 1 (with the higher number indicating better performance) *per query*. In principle, this permits us to compute the distribution of values for further error analysis, if necessary, although we only take the average over all queries for this study. In general, the NDCG and MAP are heavily correlated, although there may be some small differences.

4 Solutions

Given that LTC can be framed as an IR problem, as discussed earlier, the question arises as to *how* we solve it. However, before presenting some viable approaches for LTC, we present two reasons why it is expected to be a more challenging problem than (for example) more traditional ‘edge-discovery’ problems such as link prediction, even when given a ‘training’ taxonomy T' , or some other resource, as a ‘background’ resource.

First, while the training taxonomy is valuable and it is also domain-specific (e.g., also from the e-commerce domain, perhaps scraped from a different website), it usually has little or no overlap between C and its own concept-set, denoted as $C_{T'}$ (equivalently, $C_{T'}$ is just the set of nodes in the graph-theoretic definition of T'). Otherwise, a simple solution would have been to just induce a sub-graph on T' (over the overlapping nodes between $C_{T'}$ and the ‘target’ concept-set C over which we are trying to do local taxonomy construction) and then apply a link prediction algorithm to deduce the remaining edges. We do not discount the possibility of some overlap, but in empirical practice, the chance of overlap is minor and it is rarer still for an *edge* in the ground-truth taxonomy to overlap with an edge in a background taxonomy.⁸

Second, a challenge arises because concepts are assumed to be represented by their *labels*, which may be single words, but could also be multi-word phrases. A good method should be able to generalize to all such cases. Another important aspect to note is that, while C and $C_{T'}$ are not necessarily large, they are still too large⁹

⁸ For example, it may be that even if both taxonomies contain the concepts ‘baby clothes and toys’ and ‘diapers’, one taxonomy has an edge between them, but the other does not, meaning that the intersection would actually lead to noise when doing local taxonomy construction on the second taxonomy. The reason why this might happen is because there might be a better super-type for ‘diapers’ (e.g., ‘baby essentials’).

⁹ A general rule of thumb is several hundreds or even thousands of concepts, but not tens of thousands, of concepts.

to manually parse and build a taxonomy or ‘ontology’ (using completely manual techniques).

As intuited earlier in the introduction, we hypothesize that a more promising way (barring availability of user experiments or human-generated training data) to approach the problem is via *representation learning*, popularly known as ‘embeddings’ [24, 26]. These approaches have been used to exceed state-of-the-art performance in multiple applications, including information extraction, entity resolution and sentiment analysis [5, 21, 28]. However, no data exists on how they would perform if applied to this problem, where no examples are available. Presenting representation learning-based solutions for LTC, and demonstrating empirical applicability of such solutions, is a key contribution of this chapter.

A challenge that is specific to using embedding-based solutions for LTC is that, for structured problems similar to LTC (e.g., link prediction), the typical approach is not to use ‘natural language’ embeddings such as word2vec or BERT [4, 24], but knowledge graph (KG) embeddings like RDF2Vec and TransE [29, 36]. However, such embeddings assume the existence of a multi-relational KG where each entity has plenty of context (such as many relations and paths to other entities) to draw on, when inferring other ‘missing’ edges. The goal of these algorithms is to do prediction or completion on this semi-complete and partially noisy KG. This is also true for ‘network embeddings’ such as node2vec, DeepWalk and LINE [8, 27, 33]. A disadvantage of network embeddings that they only rely on structure and not labels. In our case, we begin with no ‘structure’ (pre-existing edges or links) and we have to only rely on labels. At the same time, there is structure in the training taxonomies that do not have label overlap, but contain domain-specific information that could be useful.

In the next section, we explore how representation learning approaches from the natural language community present a good fit for addressing these challenges, if properly adapted for LTC. We present some reasonable approaches by adapting publicly available resources in an innovative way. The solutions are empirically investigated in Sect. 5, with a qualitative discussion provided in Sect. 6.

4.1 Representation Learning Approaches for LTC

One of the questions raised in the previous section was the nature of the background resource, B . In this study, B is related to, or used by, a representation learning algorithm R . We assume that R is a word embedding algorithm, since graph embeddings are not applicable here without significant algorithmic innovation (which is not the purpose of this paper), due to ‘nodes’ in the concept-set C not having any links to begin with. The core idea is to obtain a *model* from B , either directly (such as in the case of the pre-trained embedding described below) or indirectly (by training our own embedding on a domain-specific corpus, or retrofitting an existing embedding to a related or ‘training’ taxonomy, as was mentioned earlier and is also described below). During test-time, when the concept-

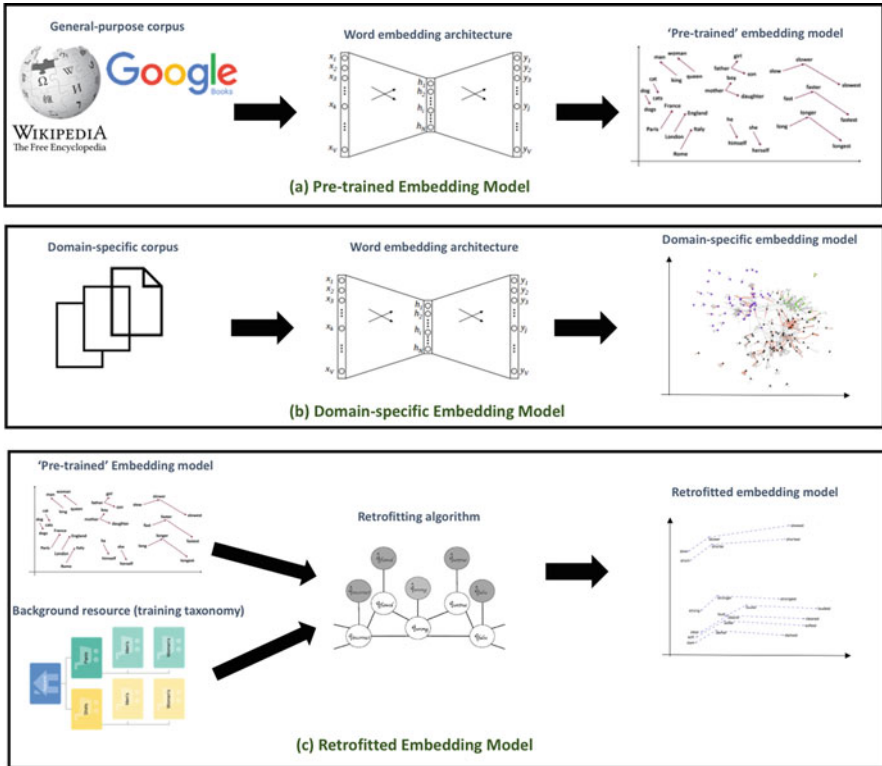


Fig. 2 A schematic showing the similarities and differences between the three solutions proposed for the LTC problem in this chapter

set C is revealed or becomes available, this model, which we also generically denote as B in a slight abuse of notation, is used to map each concept c in C to a vector \mathbf{c} , which is dense and real-valued in modern representation learning frameworks, usually comprising a few hundred dimensions. In vector space, we can then rank concepts given a query concept. Below, we first describe the different forms that B can take, followed by how to use the vectors obtained by a model for LTC. Figure 2 provides a high-level schematic of each of the proposed representation learning solutions.

Pre-Trained Embedding Model A *pre-trained embedding model* P , on which further details are provided in Sect. 5.2, is perhaps the best example of B . Such models are trained on a large (usually, but not always, openly available) corpus, such as Wikipedia and Google Books. The ‘corpora’ undergo significant preprocessing, including chunking and removal of wayward characters and punctuation, and are generally input to a word embedding algorithm such as Glove or word2vec [24, 26] as multi-sets of sequences of words. The output is a vector \mathbf{w} for word w in the

sequence. In recent years, pre-trained models have been used with great success in a variety of tasks (see e.g., [28]).

Domain-Specific Embedding Model If the domain is sufficiently different, the words in the corpus used for pre-training could carry a very different meaning (at least in the statistical sense that is relevant for word embeddings), both in theory and practice, compared to ordinary parlance. Some words simply do not show up often enough for meaningful representations to be learned (especially in legal, commercial and biomedical domains). It has become commonplace to ‘pre-train’ the embedding on a domain-specific corpus in order for it to be useful in those domains. Examples include BioBERT and Law2Vec [16, 19]. However, many domains, such as e-commerce, to which taxonomy induction tends to be applied in practice share considerable semantic overlap with Wikipedia or the Google news corpus. In part, this is expected because the users and customers in these domains are ordinary people, who may get confused if a word suddenly takes on a different meaning. Nevertheless, there are confounds, since pre-trained models may not make some important distinctions e.g., between the fruit ‘apple’ and the company ‘Apple’.

For this reason, it is also worthwhile considering (as another potential approach) an embedding trained from scratch on a domain-specific corpus D . In this formulation, the background resource is not the pre-trained embedding model P but the corpus D , which has been acquired from a domain-specific Web resource. The embedding trained on D can then be used to infer links in the same way as P (described subsequently in the *Using the Vectors for Taxonomy Induction* subsection).

Retrofitted Embedding Finally, we consider another taxonomy, called the ‘training’ taxonomy T' , as another background resource. The training taxonomy is not a ‘true’ training dataset because the concept-set on which T' was constructed is different from C . Hence, link prediction or knowledge graph embedding algorithms cannot be applied, due to the node disjointness problem (this bears resemblance to the cold-start problem observed sometimes in recommender systems [20]). Nevertheless, we assume that, by ‘retrofitting’ the pre-trained embeddings to another taxonomy from the domain, we may obtain a better embedding without re-training the whole embedding on a domain-specific corpus such as D . To this effect, we use the retrofitting algorithm implemented in [6] and traditionally used for retrofitting word embeddings to WordNet. The intuition behind that algorithm is relatively simple. For any two words in the pre-trained model P that are connected by an edge in the training taxonomy, the retrofitting algorithm’s objective function ‘pushes’ the two vectors closer together. The algorithm is allowed to run for several iterations. In this way, the final word embeddings output by the algorithm capture not only the ‘usual’ semantics of the words captured in ordinary corpora, but the domain-specific semantics embodied in the training taxonomy. For example, if ‘diapers’ and ‘baby clothes’ are neighbors (or even a short distance away) in the training taxonomy, the retrofitting will bring their embeddings closer together than would otherwise be entailed in an off-the-shelf pre-trained model.

An important detail to mention here is how we treat *unknown words* and *multi-word* phrases (such as ‘baby clothes’) in either the training or test taxonomy. When dealing with unknown words and multi-word phrases in the test taxonomy, the approach adopted is similar for all embedding methods. If the concept occurs directly as a vocabulary item in the embeddings, we use that embedding. Otherwise, we average the individual word embeddings (of words in the phrase), ignoring unknown words.

Using the Vectors for Taxonomy Induction Once the vectors have been obtained, we have to use them for the LTC task. Earlier, we proposed an IR formulation for the problem. Technical details on parameter settings that will be relevant for the empirical study are described in the next section.

Recall from Sect. 3.1 that the IR approach treats each concept $c \in C$ as a *query* q similar to keyword and other queries issued to search engines and other similar systems in the IR literature. Specifically, given a query-concept $q \in C$, where C is the concept-set over which we must induce the local taxonomy around q , we can *rank* all members of $C - q$ in descending order of each member’s *cosine similarity*¹⁰ (of its embedding) and the query embedding q . Earlier, we had also described how IR metrics, such as MAP and NDCG, can be used to evaluate the ranked list in response to a query. This is a *decentralized* way of constructing the tree, since we evaluate the method by independently computing IR metrics for each query-concept, followed by averaging.¹¹ It is well-suited to the LTC problem, though whether it can be similarly extended to the global taxonomy induction problem is for future work to address.

5 Experiments

5.1 Data

We consider three taxonomies for evaluating our approach: *Google Product Taxonomy (GPT)*, *PriceGrabber* and *Walmart*. Key statistics are provided in Table 1. The GPT is a list of thousands of ‘product categories’ designed by Google to uniformly categorize products in a shopping feed. It is publicly available at the following link¹² and has undergone some updates in recent years. We use the latest version for the experiments. PriceGrabber¹³ is a ‘smart shopping’ website that helps customers find savings and discounts on a broad category of products. The PriceGrabber taxonomy

¹⁰ Given two vectors, \mathbf{a} and \mathbf{b} , the cosine similarity is $\frac{\|\mathbf{a} \cdot \mathbf{b}\|}{\|\mathbf{a}\| \|\mathbf{b}\|}$

¹¹ Since every concept in C will be treated as a ‘query’ exactly once, the averaging will occur over $|C|$ computations of an IR metric such as the NDCG (Sect. 5.3).

¹² <https://support.google.com/merchants/answer/6324436?hl=en>.

¹³ <http://www.pricegrabber.com/>.

Table 1 Statistics on taxonomies used for the evaluation

| Name | Num. Concepts | Num. Edges | Avg. Num. Children/Node |
|--------------|---------------|------------|-------------------------|
| GPT | 5582 | 5561 | 6.36 |
| PriceGrabber | 948 | 947 | 12.62 |
| Walmart | 10,166 | 11,040 | 14.17 |

can be downloaded through an API provided to affiliate partners (merchants or publishers of product links via reviews or other useful information for potential purchasers). Walmart refers to the website¹⁴ of the well-known retailer of the same name; we obtained the taxonomy (which is technically a graph with a taxonomy-like structure) by crawling product pages starting from the sitemap.¹⁵ We extract ‘product paths’ (e.g., ‘Home/Appliances/Freezers’) from these crawled pages, and re-construct the ground-truth taxonomy from the paths.

All three of these datasets are used heavily in the real world and offer alternate perspectives of a given domain (which may be broad, such as e-commerce and online shopping, and not easily definable using formal language). The three taxonomies share some similarities such as overlap in some product categories, but differ significantly both in size and the purposes to which they have been applied in practice. This allows us to assess the effectiveness and robustness of both the representation learning methods and super-type classifier methods (described in the next section) in a controlled setting.

5.2 *Methods and Parameters*

In Sect. 4, we presented three methods that relied predominantly on representation learning, each with a slightly different rationale. We evaluate all three methods in this section to determine which one is appropriate for deriving a taxonomy given only a concept set. Note that the retrofitting method relies on an example (‘training’) taxonomy. The methods and their parameterization are briefly enumerated below. Recall that each method can be thought of as outputting a ranked list of concepts, given a query concept. Earlier, Sect. 4.1 described how we deal with phrases.

Method 1 (Pre.): Pre-trained Embeddings We downloaded the English pre-trained fastText embeddings at the following link.¹⁶ We believe that the ‘bag-of-tricks’ approach in fastText is particularly useful in our scenario (even compared to more transformer-based language models such as BERT) because its faster training (and re-training) speed allow us to perform uniform comparison between

¹⁴ <https://www.walmart.com/>.

¹⁵ The sitemap is itself a hierarchy and can be accessed at: <https://www.walmart.com/robots.txt>.

¹⁶ <https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip>.

embeddings. Also, acquiring word vectors is relatively simple, and the model is robust to misspellings and uncommon words (but which are common in specific domains). The vocabulary size of the pre-trained embeddings is 1 million words and the model was trained on the combined corpus of Wikipedia 2017, UMBC WebBase corpus and the statmt.org news dataset (16B tokens).

Method 2 (*Dom.*): Domain-Specific Corpus-Trained Embeddings We used the fastText model to obtain a domain-specific corpus-trained embedding model. We obtained the corpus as follows. First, we downloaded the *Product*-specific subset of schema.org data available on Web Data Commons.¹⁷

Although not very well known outside the Web community, schema.org has seen steady growth as structured markup embedded within HTML webpages [9]. According to the schema.org website, over 10 million sites use Schema.org to markup their web pages and email messages. Many applications from Google, Microsoft, Pinterest, Yandex and others already use these vocabularies to facilitate a rich suite of applications. The WDC schema.org project, which relies on the webpages in the Common Crawl, is able to automatically extract schema.org data from webpages due to its unique syntax and make it available as a dataset in Resource Description Framework (RDF). Since text attributes are an important part of this data, we were able to use them to construct our domain-specific corpus, as described below.

Since the *Product*-specific component of the overall schema.org data on WDC is a massive, multi-lingual corpus (more than 100GB+ in compressed form), we downloaded the first 20 chunks of the data (about 25 GB compressed), and extracted all the English language text literals from the product entities (by checking for @en and @en-US language tags). If a text literal is fully numeric, or only contains a URL (which sometimes appear within the quotes of text literals) we discard the literal. We also did standard¹⁸ pre-processing by removing unicode symbols, newlines and tabs. The final uncompressed text corpus, after cleaning and preprocessing was 5 GB, a big enough corpus to train the fastText model. We used the skipgram model (with vector dimensionality set to 300, min and max context size set to 1 and 5 respectively, and all other parameters set to their default values) for training the model. While the original corpus had 744 million words, many of these only occurred once and were spelling variations of a more frequent word. Since fastText is capable of handling such variations within the model, we took the top 1 million frequently occurring word embeddings and discarded the rest for the experiments. Before using this model, we validated its quality for taxonomy induction qualitatively by retrieving nearest-neighbors of some domain-specific words such as ‘appliance’ and ‘home’ and verifying effectiveness.

¹⁷ http://webdatacommons.org/structureddata/2018-12/stats/schema_org_subsets.html.

¹⁸ A key reference being <https://fasttext.cc/docs/en/python-module.html#important-preprocessing-data--encoding-conventions>.

Method 3 (*Retro.*): Retrofitted Embedding Transfer with Super-Type Classification Finally, we use the retrofitting approach described in Sect. 4.1. As described in that section, we retrofit the embeddings in *Pre.* using the (pre-processed) training taxonomy and the retrofitting algorithm described in [6]. We also obtain a binary *super-type classifier* (described further below) by training a random forest model. To use the super-type classifier, during the ranking phase (given a query concept), a two-step approach is adopted. First, we obtain the ranked list in descending order of cosine similarity scores, as described earlier. Next, we apply the classifier to determine the top¹⁹ 3 most likely super-type candidates for the query node and move those top 3 candidates to the top of the ranked list. In preliminary experiments, this approach was found to yield empirical benefits to standard retrofitting.

An empirical question may arise at this point: does the retrofitting even affect the pre-trained embedding significantly? To verify that it does, we computed the cosine similarity between the words' (and phrases') pre-trained embedding and the embedding achieved after retrofitting (for each of the three datasets). We then computed a frequency distribution over these cosine similarities. We only used words and phrases that occurred in the dataset, which is a limited set of all the words for which embeddings are available in the pre-trained model. Intuitively, if retrofitting had no impact on the embeddings, then the cosine similarity frequency distribution would peak at 1.0 (since the vectors would be identical both before and after the retrofitting). In contrast, if they were very different we would be seeing greater frequencies at lower cosine similarities (below 0.3). Since neither extreme is expected in practice, we should observe a distribution where there is some weight at the lower and upper extremes but much of the distribution falls between the two.

Figure 3, which plots the cosine similarity frequency distribution for the GPT dataset,²⁰ bears out this expectation. We find that retrofitting has a visible, but moderate, effect on most words' and phrases' vectors, although for some it has no effect, and for others, the effect can be marked. We found that these tend to be words like 'Apple' that have a different meaning in the e-commerce context than in the everyday context. In the next section, we explore in more detail whether retrofitting, on average, has a more positive effect on LTC performance compared to pre-trained embeddings.

We consider two 'supervised' methods that build on traditional machine learning classification and are only dependent on representation learning for their features. The idea behind the classifier is to predict the super-type of a concept node given a classifier model such as logistic regression or random forest. Since this is a supervised model, it needs to be trained, and is only applicable in the setting when a training taxonomy T' is available as background resource. Details are given below.

¹⁹ We tried multiple values for this 'reranking' parameter; 3 was found to work well across all datasets. Reranking was also found to yield superior results compared to using retrofittings without reranking.

²⁰ Distributions for the other two datasets are qualitatively similar; hence, we do not show them here.

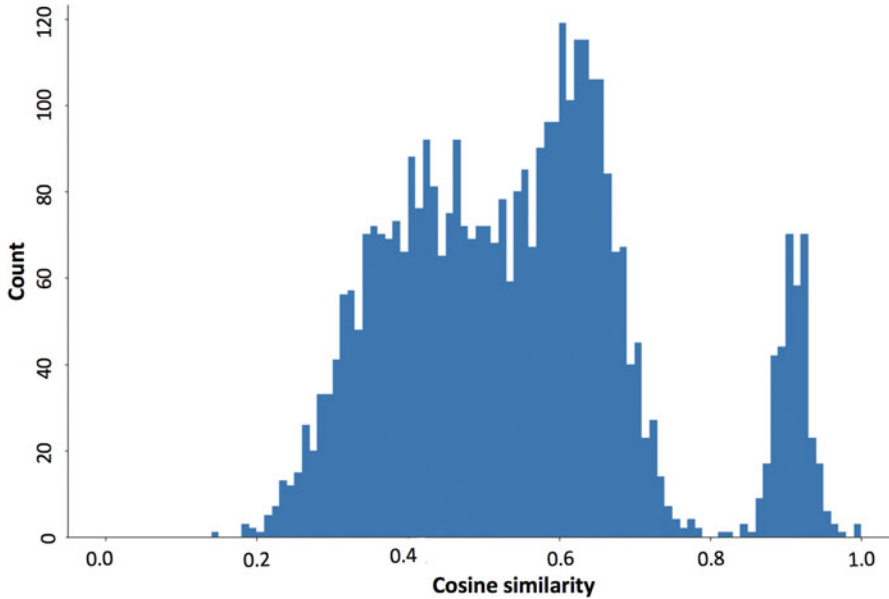


Fig. 3 A frequency distribution of the cosine similarities recorded between words' original embedding (in the pre-trained model) and the embedding obtained after retrofitting. The x -axis is the cosine similarity, and the y -axis, the counts of the cosine similarity observed (binned in narrow intervals). For this figure, we only use words and phrases that occur in the GPT data

We consider two different feature sets to fully assess the merits of the approach, described below.

Method 4 (S_{pre}): Super-Type Classification using Pre-Trained Embeddings

This model is trained using the pre-trained embeddings (*Pre.*) as features. We use a random forest model with default parameters, but with number of estimators set to 200. We frame the problem as binary classification, using the difference of the embedding between the query node and a candidate node as the 'feature vector'. To train the model, we use the training taxonomy T' . Specifically, for each node c_i in T' that has a parent c_j (recall that there might be a set of upper-level nodes that may not have a parent), we construct the *translational* feature vector $\mathbf{c}_j - \mathbf{c}_i$ and assign it the positive label. We randomly sample a sibling of c_i (say, c_k) and assign the feature vector $\mathbf{c}_k - \mathbf{c}_i$ the negative label, thereby obtaining a balanced training set, using only concepts in T' . We obtain the full ranked list on the actual (i.e. the test) concept set C by ranking all candidates in $C - q$ (q being the query) according to the probability scores output by the classifier (higher probability indicating higher likelihood of that concept being a super-type of q).

Method 5 (S_{retro}): Super-type Classification using Retrofitted Embeddings

This model is identical to the above, except that we use the embeddings obtained

from Method 3 (*Retro.*) for deriving the features. This baseline also offers us the opportunity to study the difference between this method and Method 4 in a second-order context (where the embeddings are themselves input to a model), rather than used directly (via cosine similarity) for obtaining the rankings. Note that this model is also used in the second step of the two-step model previously described for *Retro.*

5.3 Methodology and Metrics

We stated in Sect. 3.1 that an Information Retrieval (IR)-based methodology could be used to evaluate the effectiveness of any ranking-based approach to local taxonomy construction. In the IR methodology, we treat each concept in the concept-set C as a *query*, and compute a ranking over all *other* concepts in the concept-set. All the methods described in the previous section can be tuned to return a ranking of all concepts, given a concept. Since we know the ground-truth taxonomy for the concept-set, we know which concepts in the ranking (for each ‘query’ concept) are ‘relevant’ (to use IR terminology); these are precisely the concepts that are neighbors of the query concept in the ground-truth taxonomy.

We have already presented two viable IR metrics earlier (NDCG and MAP). We use these metrics to evaluate performance on the three datasets, using all the methods presented in the previous section. However, one concern that we had raised in the earlier section is that, while IR may be a good framework for LTC, it may not be completely satisfactory for global taxonomy induction. While global taxonomy induction is not the primary subject of this work, it is an interesting issue to understand how ‘far off’ we are in achieving good global taxonomy induction performance. To that end, we describe an additional *structural* tree-based metric below that is better suited for assessing the global quality of an induced tree. We use this set of results to provide some more insight on the difficulty of global taxonomy induction compared to LTC.

Before we can use the tree-based metric, we note that it requires two tree-like taxonomies as input, one of which is the ground-truth taxonomy. To induce a full taxonomy over a concept-set $|C|$, we consider the following simple approach. First, we construct an approximate k -regular graph, which is a graph where each node has exactly k neighbors. We do this by mapping each concept to a node, and then connecting a node to the top- k nodes in the ranked list retrieved by the approach. Next, we execute a classic (i.e. Kruskal’s) *maximum spanning tree* (MST) algorithm over this approximate²¹ k -regular graph [17].

²¹ One might wonder why the graph is only approximately k -regular if we always consider the top- k neighbors of a node. The reason is that each node is considered as a query in turn, and ranking can be asymmetric. For example, suppose we used $k = 4$ and the neighbors of a node c_1 were c_2, c_3, c_4, c_5 . Hence, c_1 would initially have 4 neighbors. However, if there were some other node

The ‘ideal’ metric for comparing two taxonomies is the tree edit distance [2]. However, because of its time complexity, tree edit distance implementations cannot handle taxonomies with more than a few hundred nodes. In contrast, two of the datasets presented in Table 1 have thousands of nodes. For this reason, rather than use tree edit distance, we consider another method of robustly estimating the similarity between two taxonomies based on *shortest path overlap*. The metric (again between 0.0 and 1.0, with higher values indicating greater similarity between an output taxonomy and the ground-truth taxonomy, and hence indicating better performance of the approach) is calculated as follows. First, we randomly sample 20% of the concepts in the test dataset. We call this set the *source concept set* $S_C \subseteq C$, where C is the concept-set. We compute the shortest path between each concept c (equivalently, taxonomy node) in S_C and $C - c$, representing the nodes in the shortest path as a set rather than a sequence. This yields a ‘bag’ (or multi-set) of sets of concepts for the ground-truth (G) and for each algorithm’s output (O). Designate these (bags) respectively as G_S and O_S , we can compute the Jaccard similarity $J = \frac{|G_S \cap O_S|}{|G_S \cup O_S|}$ between the two bags.

$$J = \frac{|G_S \cap O_S|}{|G_S \cup O_S|} \quad (3)$$

We note again that both the IR methodology and the structural methodology have their uses, albeit in different applications and circumstances. The IR metrics are useful when the goal is not to recover the entire tree, but to treat each concept as a query and to get relevant or related concepts (e.g., when building related-product recommenders). The direct comparison of taxonomies is more useful if one is interested in recovering a taxonomy that is, in its global structure, very similar to the test taxonomy. The structural metric is biased more heavily towards the test taxonomy, which may make it less favorable for domains and concept-sets that are inherently under-determined (i.e where more than one taxonomy with globally varying structure has high quality).

5.4 Results

Table 2 contains the results for all combinations of training/test taxonomies using the IR-based methodology described in the previous section. From the table, we find that the maximum NDCG/MAP scores for each train/test setting always corresponds to one of the representation learning algorithms, as opposed to the super-type classifiers. This shows that, at least on a per-query basis, direct use of embeddings is a better choice than using the embeddings in a downstream machine learning model.

(say, c_{10}) that had c_1 as one of its neighbors, then c_1 would have (at least) 5 neighbors. The same observation could apply to any node.

Table 2 Information retrieval metrics (NDCG/MAP) on each of the six train/test settings. Except for S_{pre} and $Retro.$ on the Walmart/PriceGrabber case (last row), all results are significant at the 95% level. Furthermore, except for $Dom.$ (GPT/PriceGrabber, Walmart/PriceGrabber), all results are significant at the 99% level. Maximum values at the row-level for each of the two metrics are in **bold**. Since not all methods use the ‘training’ taxonomy, some results may be repeated

| Train | Test | $Retro.$ | $Pre.$ | $Dom.$ | S_{pre} | S_{retro} |
|--------------|--------------|------------------|------------------|------------------|------------|-------------|
| GPT | PriceGrabber | 0.44/0.30 | 0.38/0.23 | 0.37/0.22 | 0.26/0.10 | 0.34/0.18 |
| GPT | Walmart | 0.27/0.13 | 0.29/0.16 | 0.29/0.15 | 0.14/0.023 | 0.19/0.07 |
| PriceGrabber | GPT | 0.34/0.18 | 0.38/0.23 | 0.37/0.21 | 0.17/0.04 | 0.23/0.096 |
| PriceGrabber | Walmart | 0.25/0.11 | 0.29/0.16 | 0.29/0.15 | 0.13/0.017 | 0.18/0.056 |
| Walmart | GPT | 0.35/0.20 | 0.38/0.23 | 0.37/0.21 | 0.16/0.03 | 0.29/0.156 |
| Walmart | PriceGrabber | 0.39/0.25 | 0.38/0.23 | 0.37/0.22 | 0.23/0.07 | 0.38/0.23 |

Reinforcing this conclusion, the rank correlation between MAP and NDCG is also high i.e. if we rank the five approaches either by MAP or by NDCG we get very similar results.

Importantly, we observe that the retrofitting embedding $Retro.$ tends to outperform $Pre.$ when the training taxonomy is large (as in the case of Walmart) and when the testing taxonomy is small (as in the case of PriceGrabber). However, noise also plays a role, since the GPT taxonomy is (arguably) the ‘cleanest’ of all the taxonomies, having been meticulously designed at Google with several iterations over the years. We hypothesize that, for this reason, the best performance achieved by $Retro.$ is in the GPT/PriceGrabber setting rather than the Walmart/PriceGrabber setting, although it also does quite well (both absolutely and relatively) on the latter.

An overall comment that we make with respect to these results is that the numbers clearly show that there is usually at least one relevant entry in the top 3. Typically, this is necessary to achieve an NDCG or MAP of greater than 33%. This is encouraging, since it hints towards the feasibility of the problem. Furthermore, the inferior performance of $Dom.$ compared to the other representation learning models suggests that that pre-trained embeddings are sufficient, and that it is not necessary to invest significant amounts of time in preprocessing and preparing domain-specific corpora for re-training embedding models.

Next, we turn to results on the ‘structural’ tree-based metric we had described earlier as a means for evaluating the quality of a global taxonomy that is induced by merging LTC outputs and applying the classic MST algorithm. Table 3 tabulates these results. Unlike the IR-based results, the results in Table 3 demonstrate that super-type classification, using pre-trained embeddings, is more useful for capturing the overall structure of the taxonomy. The results in Table 3 are also far more uniform than the results in Table 2. This is strikingly true both when we consider the row-wise distribution (how different approaches perform with respect to a given training/test taxonomy) and the column-wise distribution (how a single approach performs across different training/test taxonomies).

One question that may arise is whether the numbers can be further improved by incorporating cosine-similarity weights in the construction of the k -regular graph

Table 3 Average Jaccard similarities between shortest path sets (of output and ground-truth) as described in the text (weighted/unweighted MST). Other than S_{pre} for Walmart/PriceGrabber case (for which, $0.01 < p < 0.05$), all results are significant at the 99% level. Maximum values at the row-level for each of the two metrics are in **bold**

| Train | Test | <i>Retro</i> | <i>Pre</i> | <i>Dom</i> | S_{pre} | S_{retro} |
|--------------|--------------|--------------|------------|------------|------------------|-------------------|
| GPT | PriceGrabber | 0.14/0.22 | 0.13/0.21 | 0.14/0.19 | 0.22/0.23 | 0.18/ 0.23 |
| GPT | Walmart | 0.11/0.18 | 0.10/0.17 | 0.092/0.16 | 0.19/0.22 | 0.13/0.18 |
| PriceGrabber | GPT | 0.16/0.21 | 0.14/0.19 | 0.11/0.17 | 0.23/0.22 | 0.21/ 0.22 |
| PriceGrabber | Walmart | 0.11/0.18 | 0.10/0.17 | 0.09/0.16 | 0.27/0.24 | 0.16/0.22 |
| Walmart | GPT | 0.13/0.18 | 0.14/0.19 | 0.11/0.17 | 0.22/0.22 | 0.18/0.21 |
| Walmart | PriceGrabber | 0.14/0.21 | 0.13/0.21 | 0.14/0.19 | 0.22/0.21 | 0.16/ 0.21 |

described in the previous section. The intuition is that, if we incorporate those weights, it may lead Kruskal’s spanning tree algorithm to produce a better tree.

However, in comparing the weighted versus unweighted results in Table 3, we find that there is hardly any difference for the super-type classifiers (in fact, the unweighted performance is better). However, unequivocally, for the representation learning approaches, the *unweighted* MST performs much better, illustrating that the raw weights may add noise, even though their retrieval performance is good. This further suggests that it is more fruitful to use these embeddings in a ranking-based setting. Additional techniques may be necessary to convert LTC results into a global taxonomy.

Furthermore, the results in Table 3 should be assumed to be pessimistic, since there is more than one ‘right’ way to construct a potential taxonomy from the concept-set, whereas we have conservatively assumed that the ground-truth taxonomy is the only ‘right’ way. Human-centric evaluation may be necessary to assess true performance of each method, but the cost may be prohibitive. An alternative could be to annotate sampled edges in the taxonomy, but this would only have local validity. Effectiveness of the induced taxonomy could also be indirectly measured on a downstream task. We leave a direct comparison of these evaluation choices for future research, as the main subject of the research in this article is LTC.

6 Discussion

While the results in Sect. 5.4 provide some insight into the performance of the representation learning algorithms using a variety of metrics on the LTC problem, we provide a brief qualitative assessment in this section to understand those results at a more intuitive level. Figures 4, 5, and 6 respectively provide an example from PriceGrabber, GPT and Walmart, respectively. In order to retrieve a consistent taxonomy, we took the additional step of running the MST algorithm (as described in the previous section for computing the structural tree-based metric) and we show, for a selected query in each of the three datasets, the ‘parent’ concept of that query

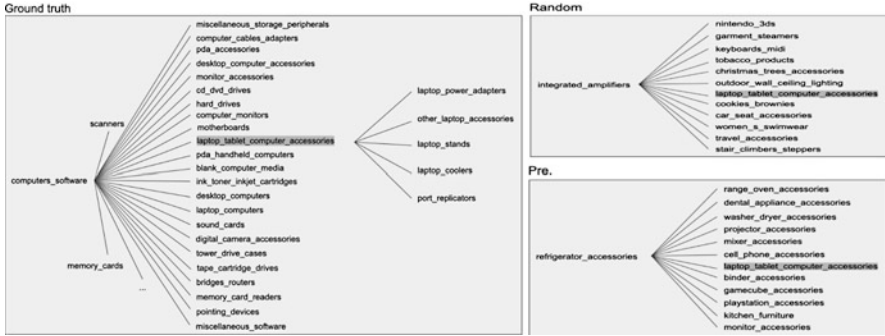


Fig. 4 Fragments of the ground-truth taxonomy, random retrieval and retrieval using the pre-trained baseline (with *laptop_tablet_computer_accessories* as the query) from the PriceGrabber dataset

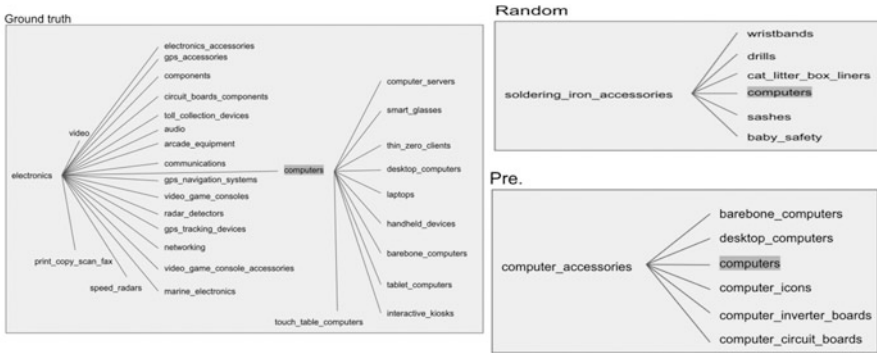


Fig. 5 Fragments of the ground-truth taxonomy, random retrieval and retrieval using the pre-trained baseline (with *computers* as the query) from the GPT dataset

concept in the tree output by the MST, as well as the *other* children of that parent concept. Because we took this additional step, the qualitative assessment in this section should not be seen as an assessment of LTC itself, but of the more advanced version of the problem where we attempt to induce a global taxonomy from the LTC results.

In the case of PriceGrabber, we use the (randomly selected) query *laptop_tablet_computer_accessories* for illustrative purposes. We show both the ground-truth neighborhood, as well as the results achieved when using a random baseline (for each query concept, we randomly rank the other concepts and then perform the MST step) and the *Pre.* baseline. The result shows why the problem is a difficult one. Despite surface similarities between the parent concept (*refrigerator_accessories*) and the query, semantically, the former is not related to the latter. Results in the LTC version of the problem were better, but when a global taxonomy was induced over the LTC results using the MST algorithm, the performance declined.

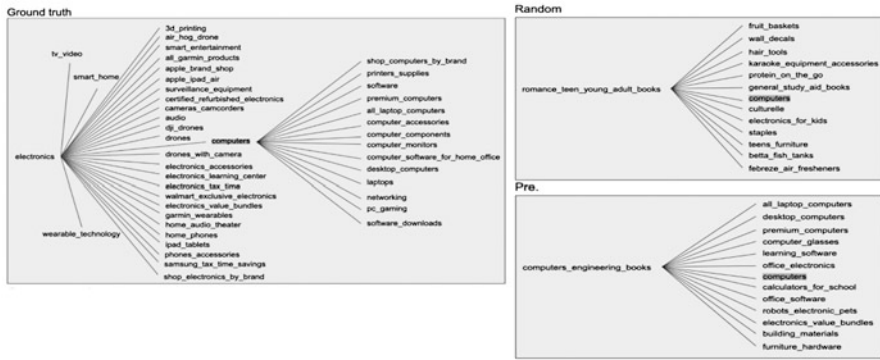


Fig. 6 Fragments of the ground-truth taxonomy, random retrieval and retrieval using the pre-trained baseline (with *computers* as the query) from the Walmart dataset

Results are moderately better for GPT; while ‘computer_accessories’ is clearly a relevant entry for ‘computer’, it should not have been a parent concept, since computer_accessories is a sub-category of ‘computer’, not the other way around. Unlike the PriceGrabber case, this is a positive result from the view of LTC, where we are only looking to retrieve relevant results (both parent and children) rather than the directed structure of the global taxonomy. However, it is a negative result from the view of the global taxonomy induction problem. The example highlights an avenue for future research; namely, devising good algorithms for global taxonomy induction. A similar observation applies to Walmart.

7 Potential Applications

As intuited in the *Introduction* section, LTC can enable several applications in enterprise. We list a few possibilities below. While some of these possibilities are already being realized, others remain to be investigated:

- There is already some work linking taxonomy induction to building better recommender systems [3, 38], but much more work is possible by integrating LTC directly into recommendation algorithms. Also, much of the work seems to have been done in the research setting. We are not aware of a synergy between recommenders and LTC solutions in industry, which would be a valuable potential application for companies looking to get a competitive advantage.
- Due to the close connection between IR and LTC, it may be possible to use LTC to improve the search experience on websites and other user-facing portals. Compared to the advanced search algorithms implemented by major companies like Google, search experience on individual websites is not as impressive. Integrating LTC into the website’s search engine may provide valuable benefits.

- We believe there is considerable scope for knowledge engineers and domain experts to use LTC solutions before commencing on an ontology design process. However, for this to occur at scale, the LTC algorithms would have to be ‘wrapped’ in user-friendly software that can be easily deployed and used by domain experts. LTC could also be offered as a ‘service’ on the Web (e.g., software as a service, or SaaS). For example, a user could just upload their concepts as a text file to the service, and a taxonomy would be returned to the user. Such a service could be easily monetized if the quality of induction is high enough, and eventually, global taxonomy induction solutions could also be incorporated into the service. To the best of our knowledge, no such service is currently available.
- Rendering the output of LTC as a tree or graph may allow users to interactively visualize, explore and manipulate large concept-sets. In our own group, we are currently working on developing such a customizable visualization tool.

8 Future Work

Due to the potential applications of LTC, as described in Sect. 7, the growth of relevant application domains such as e-commerce, as well as the relative novelty of the problem compared to many other problems in AI, there are significant opportunities for future research. Below, we describe some of the key future research areas we have identified:

- An important avenue of future work is to develop methodologies and algorithms that are applicable to the *global taxonomy induction* problem. As we stated earlier, this problem is very different from standard link prediction and triples classification problems (in the knowledge graph community) due to the complete lack of initial structure or network. In fact, it is not even completely clear how one would evaluate such an induced taxonomy with respect to a ‘ground-truth’ taxonomy. IR cannot be applied in an obvious way, unlike the LTC problem described here, and neither can measures like tree edit distance and graph edit distance due to their quadratic time and space complexity (in the number of nodes). Hence, new metrics, such as the shortest paths overlap metric presented earlier, may be required to evaluate these algorithms. The strengths and limitations of these ‘cheaper’ metrics (especially compared to tree edit distance) are not well understood, however.
- A second promising opportunity is to expand the techniques in this paper for domains other than e-commerce. For example, taxonomies and concept-sets are widely used in domains like medicine. We suspect that, in such specialized domains, the domain-specific embedding model might end up achieving better performance than the pre-trained embedding model, unlike the results in this paper. However, this hypothesis would need to be investigated in future experiments.

- Theoretically, it would be fruitful to consider whether IR is the optimal way of modeling this problem, or if there are other frameworks (such as a graph-theoretic formalism) where the algorithms and metrics allow us to address the challenges more effectively. An advantage of using graph-theoretic approaches is that they may be uniformly applicable both to the local and the global versions of the problem, which allows us to compare the difficulty of both problems from a common algorithmic standpoint.
- On a more practical front, and in line with what we described earlier in Sect. 7, it is important to move beyond research and aim to *apply* the research toward practical tooling. Developing taxonomy induction ‘modules’, for example, that operate on the basis of the algorithms presented in this paper, and that can be integrated into existing software workflows, may yield significant benefits for an enterprise.
- Finally, investigating the scalability of the approach and comparing to more novel research in the NLP community (especially, transformer-based models such as BERT and RoBERTa) may be a valuable area of research for those looking to improve the quality of the initial set of representation learning-based systems presented in this paper.

9 Conclusion

Local taxonomy construction, or LTC, is the problem of locally inducing a concept’s neighborhood, given a pre-existing set of ‘target’ concepts, without being given any example links. While it has many practical applications (Sect. 7), the problem has witnessed relatively little research attention compared to similar problems such as link prediction. LTC is a more tractable version of the global taxonomy induction problem, but still significantly more difficult than ordinary link prediction.

This paper formalized the LTC problem using concepts from Information Retrieval (IR). In a detailed empirical study, we also presented and evaluated solutions based on representation learning algorithms, derived and adapted from the NLP community. These algorithms were found to be viable solutions, but the experimental results collected using three real-world datasets in the e-commerce domain also indicate that there is considerable room for improvement. Finally, we identified several potential applications and promising areas of future research.

Acknowledgments We gratefully acknowledge funding under the Yahoo! Faculty Research Engagement Program.

References

1. Albaum G (1997) The likert scale revisited. *J Mark Res Soc* 39(2):1–21
2. Bille P (2005) A survey on tree edit distance and related problems. *Theor Comput Sci* 337(1–3):217–239
3. Cho YH, Kim JK (2004) Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert systems with Applications* 26(2):233–246
4. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*
5. Ebraheem M, Thirumuruganathan S, Joty S, Ouzzani M, Tang N (2018) Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment* 11(11):1454–1467
6. Faruqui M, Dodge J, Jauhar SK, Dyer C, Hovy E, Smith NA (2014) Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*
7. Fensel D (2001) Ontologies. In: *Ontologies*. Springer, Berlin, pp. 11–18
8. Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp 855–864
9. Guha RV, Brickley D, Macbeth S (2016) Schema. org: evolution of structured data on the web. *Commun ACM* 59(2):44–51
10. Gupta A, Lebrecht R, Harkous H, Aberer K (2017) Taxonomy induction using hypernym subsequences. In: *Proceedings of the 2017 ACM conference on information and knowledge management*, pp 1329–1338
11. Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*
12. Kejriwal M (2019) *Domain-specific knowledge graph construction*. Springer, Berlin
13. Kejriwal M (2021) A meta-engine for building domain-specific search engines. *Software Impacts* 7:100052
14. Kejriwal M, Knoblock CA, Szekely P (2021) *Knowledge Graphs: Fundamentals, Techniques, and Applications*. MIT Press, New York
15. Kejriwal M, Selvam RK, Ni CC, Torzec N (2020) Locally constructing product taxonomies from scratch using representation learning. In: *2020 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*. IEEE, New York, pp 507–514
16. Kim N, Kim HJ (2017) A study on the law2vec model for searching related law. *Journal of Digital Contents Society* 18(7):1419–1425
17. Kruskal JB (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc Am Math Soc* 7(1):48–50
18. Lancichinetti A, Fortunato S (2009) Community detection algorithms: a comparative analysis. *Phys Rev E* 80(5):056117
19. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020) Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36(4):1234–1240
20. Lika B, Kolomvatsos K, Hadjiefthymiades S (2014) Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41(4):2065–2073
21. Liu L, Ren X, Zhu Q, Zhi S, Gui H, Ji H, Han J (2017) Heterogeneous supervision for relation extraction: A representation learning approach. *arXiv preprint arXiv:1707.00166*
22. Manning CD, Raghavan P, Schütze H (2008) *Introduction to information retrieval*. Cambridge University Press, Cambridge
23. Mao Y, Ren X, Shen J, Gu X, Han J (2018) End-to-end reinforcement learning for automatic taxonomy induction. *arXiv preprint arXiv:1805.04044*
24. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, pp 3111–3119

25. Mrkšić N, Séaghdha DO, Thomson B, Gašić M, Rojas-Barahona L, Su PH, Vandyke D, Wen TH, Young S (2016) Counter-fitting word vectors to linguistic constraints. arXiv preprint arXiv:1603.00892
26. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
27. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 701–710
28. Rezaeinia SM, Rahmani R, Ghodsi A, Veisi H (2019) Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications* 117:139–147
29. Ristoski P, Paulheim H (2016) Rdf2vec: Rdf graph embeddings for data mining. In: *International Semantic Web Conference*. Springer, Berlin, pp 498–514
30. Sears A, Jacko JA (2009) *Human-computer interaction fundamentals*. CRC Press, New York
31. Singhal A (2012) Introducing the knowledge graph: things, not strings. *Official google blog* 5:16
32. Snow R, Jurafsky D, Ng AY (2006) Semantic taxonomy induction from heterogenous evidence. In: Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics. Association for Computational Linguistics, New York, pp 801–808
33. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web, pp 1067–1077
34. Velardi P, Faralli S, Navigli R (2013) Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Comput Linguist* 39(3):665–707
35. Wang P, Xu B, Wu Y, Zhou X (2015) Link prediction in social networks: the state-of-the-art. *Science China Information Sciences* 58(1):1–38
36. Wang Q, Mao Z, Wang B, Guo L (2017) Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans Knowl Data Eng* 29(12):2724–2743
37. Xian Y, Lampert CH, Schiele B, Akata Z (2018) Zero-shot learning a comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans Pattern Anal Mach Intell* 41(9):2251–2265
38. Zhang Y, Ahmed A, Josifovski V, Smola A (2014) Taxonomy discovery for personalized recommendation. In: Proceedings of the 7th ACM international conference on Web search and data mining, pp 243–252
39. Zhuang F, Cheng X, Luo P, Pan SJ, He Q (2015) Supervised representation learning: transfer learning with deep autoencoders. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*

The Evolution of Online Sentiments Across Italy During First and Second Wave of the COVID-19 Pandemic



Francesco Scotti , Davide Magnanimiti, Valeria Maria Urbano, and Francesco Pierri

Abstract During crises, online social networks represent a primary source of information to understand the opinion and feeling of individuals. In this paper, we analyse Twitter conversations during both waves of COVID-19 pandemic in Italy (respectively March–April 2020 and October–December 2020) and shed light on the main factors which drive online sentiment.

Through the application of cross-section and panel regression models, we find that, during the first wave, more negative feelings are expressed online by users residing in provinces with higher contagions and larger extra mortality rates. During the second wave, conversely, we observe that online sentiment is mainly affected by socio-economic variables since more positive sentiment is associated to users located in areas with larger income per capita, deprivation index and propensity to telework. Over the entire period of observation, we detect a stronger improvement in online sentiment of users located in the provinces located in the North of Italy, which is characterized by larger wealth and better healthcare infrastructures.

Keywords COVID-19 · Regression · Sentiment analysis · Twitter

1 Introduction

As the pandemic of SARS-COV-2 was spreading around the globe during 2020, people witnessed a surge of public interest around social media data as a vital source of information to counter the virus [10]. As a matter of fact, monitoring human activity on social network may enable authorities and policy makers to detect

F. Scotti (✉) · V. M. Urbano

Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Milan, Italy

e-mail: francesco.scotti@polimi.it; valeriamaria.urbano@polimi.it

D. Magnanimiti · F. Pierri

Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy

e-mail: davide.magnanimiti@polimi.it; francesco.pierri@polimi.it

outbreaks at an early stage, minimize their spread [6, 10, 21, 31] and understand the evolution of individuals feelings and opinions during such an unprecedented crisis [1, 4, 17, 25, 36].

The usage of textual analytics for identifying public sentiment, indeed, provides relevant insights on attitude, perceptions and behaviours of individuals. Large scale extractions of people sentiment from social media may support policy makers and healthcare organizations assessing the needs of citizens in real time and designing targeted communication strategy. Although several research studies focused on the analysis of individuals' feelings and opinions through social media data emphasizing the role of sentiment analysis, the factors associated with positive and negative sentiments of people were barely investigated. The combination of sentiment analytics with socio-economic variables of diverse areas may create insights contributing to the development of a clear understanding of factors that drives individual's feelings [26].

In this spirit, this paper aims at investigating the factors affecting the feelings of Italians during the COVID-19 pandemic. Specifically, we combine Twitter data with socio-economic characteristics and epidemiological variables of administrative units to understand whether user sentiments were driven by socio-economic and epidemiological factors. We significantly consolidate a preliminary analysis of online sentiment expressed by Italian users on Twitter during the first wave of COVID-19 pandemic in Italy (February–April 2020) [30], by extending the period of observation to include the second wave of contagion and additional covariates in the regression model. Moreover, we perform an analysis of the factors driving the change in the online sentiment between the two periods.

First and foremost, this research study provides relevant insights on the impact of socio-economic characteristics and the severity of coronavirus spread on users' feelings during the two waves of contagion. Secondly, our analysis investigates the evolution of sentiments across time and analyse the factors driving the change in users' sentiments. To pursue our research goal, we perform an automatic extraction of tweets sentiment and deployed cross-section and panel regression models to understand the interplay between feelings expressed by online users and a set of socio-economic and epidemiological variables, computed at province level. Results show that during the first period sentiments are mainly driven by the severity of contagion, revealing that individuals perceived COVID-19 as an health emergency. On the opposite, during the second wave socio-economic factors, such as local wealth and level of disparities mainly affect users' feelings.

The outline of remaining sections is as follows: in the next section we describe research studies related to the use of sentiment analysis for analysing social media data during epidemics. We then provide a brief description of the datasets used in our analysis, and we present our methodology. Finally, we discuss results of the analysis, and we draw conclusions and future directions for our work.

2 Related Literature

The proliferation of social media usage for sharing opinions, facts and feelings by individuals has provided unprecedented opportunities to analyse communication patterns and sentiment spread across communities. In the public health domain, Twitter data have been used for surveillance activities [19, 21] but also for understanding public attitudes and behaviours during past epidemics such as the N1H1 influenza, Ebola, Zika and Mers-Cov. Monitoring feelings of individuals through social media data may, indeed, support authorities in responding to public concern and developing appropriate communication strategy.

During the 2009 N1H1 epidemic, a content analysis of Twitter data revealed the importance of the platform as a rich source of opinions and personal experiences shared by individuals, which could be used to support health authorities to better respond to public concerns [12]. In 2010, several sentiment classification methods allowed to detect N1H1-related message from Twitter and compare results with official statistics from the Centers for Disease Control and Prevention (CDC) obtaining an 84.29% accuracy in classifying messages [14]. In 2015, the use of sentiment analysis and count based techniques of social media data were introduced for the detection of N1H1 symptoms in India based on Twitter data [18]. A similar analysis discussed the effects of MERS-CoV infection on the opinions of individuals in Saudi Arabia [37]. The study suggested the usage of automated systems for analysing public opinions in order to help decisions makers to face health emergencies.

During the Ebola outbreaks, several research studies focused on analysing sentiments and feelings of Twitter users. In 2014, the epidemic provoked mainly negative emotions such as anxiety, anger, swearing and death [16], while in another research study conducted in 2015, researchers confirmed that panic was the prevalent sentiment in the general landscape of social media [32]. In the same year, a medical health information system, named 'eMood', started to collect social media data and visualized the results of the sentiment analysis [13]. Besides providing an overview on people's feelings during the Ebola spread, the tool developed by researchers was able to identify most influential users providing relevant insights on users' network of relationships and influence patterns.

Similar lines of research have been followed during the recent pandemic of the novel COVID-19 disease. In an observational study [22], the authors described how Twitter content and sentiment evolved in the early stages of the COVID-19 outbreak. Results showed that tweets were predominantly associated with negative sentiment such as fear, surprise and anger. Another sentiment analysis performed in the same period revealed the presence of contrasting emotions among citizens as joy and sadness, disgust and surprise [20]. This may reveal the absence of a homogeneous perception of the severity of the situation, and explain the difficulties experienced by policy makers and health authorities to transmit a harmonized and coherent message to society. This issue was corroborated by some evidence exhibiting that individuals on Twitter relied more upon common users generated content rather

than on official government and other related institutional communications [29]. Sentiment of individuals has also been analysed through the lens of epidemic psychology. Studying the use of language across 39M tweets shared in the US, scholars identified three different phases characterizing online reactions to the pandemic: a refusal stage, while only other countries were experiencing deaths, a suspended reality period, which started after the first COVID-19 victim was announced in each state and, finally, an acceptance phase, when authorities imposed social distancing measures, in which individuals found a “new normality” for their daily activities, conveying more positive sentiments [1].

Despite the large amount of scientific works which covered the impact of epidemics on individuals’ feelings, to the best of our knowledge there is only a handful of recent contributions that analysed to what extent the swing of emotions is influenced by the socio-economic conditions of the population across different geographical areas. In 2017, a study evaluated the spread of cholera in Uganda, investigating socio-economic characteristics of population to understand whether the promotion of social services, such as education, could contribute to control the virus [11]. During the current pandemic, it was carried out an analysis on the association between socio-economic variables and sentiment of US citizens about lifting mobility restriction put in place to cope with the COVID-19 health crisis [26]. They showed that family households, individuals with limited education levels, low-income and higher house rent are more interested in restarting the economy, thus providing relevant signals to policymakers that seek to reduce the impact of the pandemic.

3 Data Collection and Description

3.1 *Twitter*

To investigate online sentiment of Italian individuals we focus on Twitter conversations about COVID-19, and we employ two different collections of tweets shared during both waves of the pandemic in Italy. The first one is part of a collection of messages from Twitter in the Italian language that is continuously going on since 2012 at the University of Turin [3]. tweets in this dataset are exclusively Italian and filtered with specific keywords related to the COVID-19 pandemic (e.g. covid, coronavirus, quarantena, iorestocasa, etc.). We further collected additional data by leveraging the Twitter Streaming API to collect Italian tweets using the same set of keywords. The final resulting collection of tweets is composed by approximately 1.7 M tweets for the first wave and almost 800 K for the second wave of the pandemic, corresponding respectively to the periods February 17th–May 5th and October 1st–December 31st. We finally geo-located users based on the location disclosed in their Twitter profile (when available) and matching it against a list of 8 k Italian municipalities. Overall, we were able to geo-locate 400 K tweets during the first

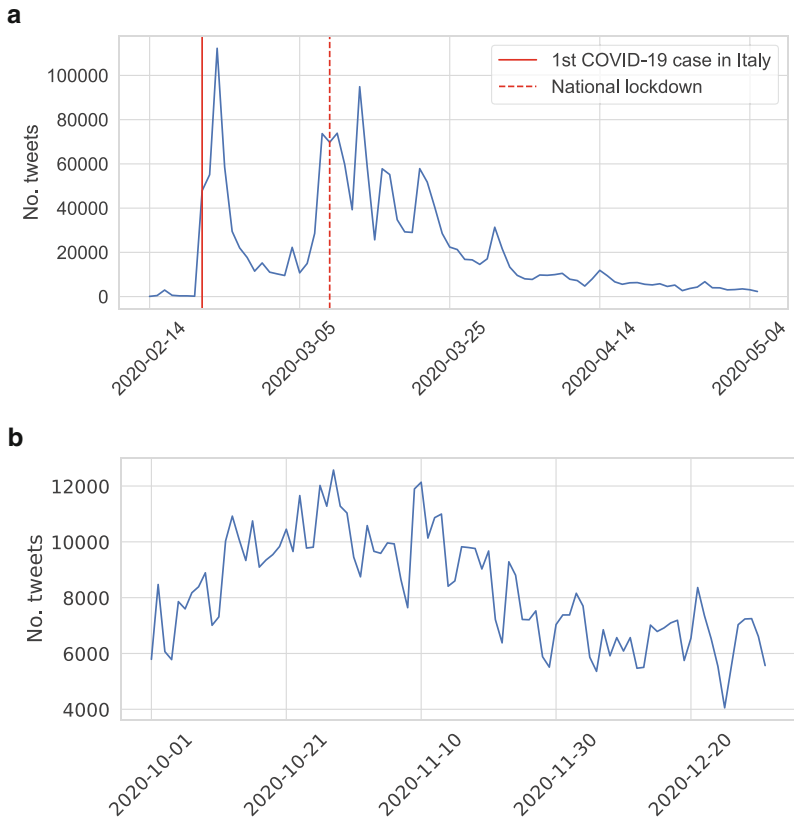


Fig. 1 Number of daily tweets during the first (a) and the second (b) COVID-19 wave periods. Red lines indicate respectively the 1st diagnosed COVID-19 case in Italy (solid) and the beginning of national lockdown (dashed)

period and 260 K tweets during the second one. Figure 1 shows the time series of the daily number of tweets during the two distinct periods of observations. We can notice a striking difference in the number of tweets shared during the two periods, with strong peaks in correspondence of the COVID-19 outbreak in Italy (February 21st) and the following national lockdown (March 9th).

3.2 Socio-Economic and Epidemiological Variables

In order to investigate the main factors affecting online sentiment of Italian individuals we consider several socio-economic variables that measure local wealth, level of disparities and social cohesion of territories, as well as the possibility of working remotely (see Fig. 2 for the geographical distribution of these variables).

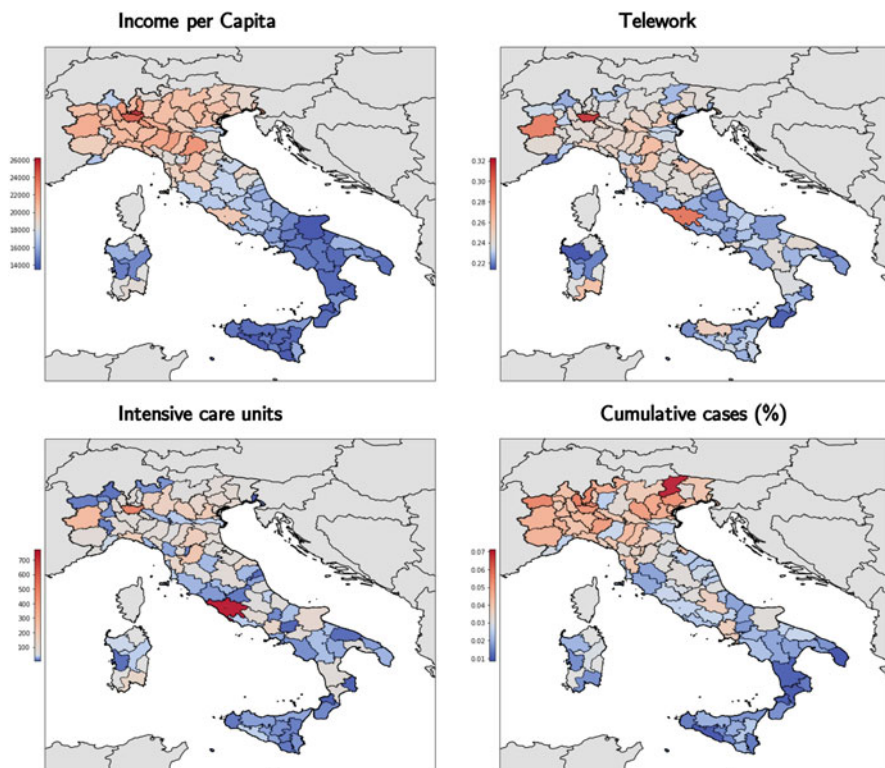


Fig. 2 Geographical distribution of several socio-economic variables at province level. The colorbar is centered at the median value of the distribution

In order to measure local wealth, we take into account the *Average income per capita* and the *Fiscal Capacity*, which indicate respectively the average income declared by taxpayers in 2018 in the underlying province and the administrative revenues based on property tax, local income tax, and local fees (see Table 1 for more details on the sources of variables and the year of reference).

Secondly, as a proxy of the level of disparities we include in the model the *Inequality* index, disclosed by the Ministry of Economics and Finance (MEF) and computed as the ratio between mean and median values of the distribution of declared income in 2018. In this case, provinces characterized by values larger than 1 are those where wealth is less equally distributed, since the distribution of income displays a positive skewness which is a symptom of a few individuals with large income and a larger concentration of citizens with lower wages. Moreover, to measure social cohesion of territories we consider the *Deprivation* index,¹ which is a composite indicator covering the dimensions of education, unemployment, housing,

¹ Data refer to the year 2016.

Table 1 Source and Year of reference for the socio-economic and epidemiological variables

| | Source | Year |
|---------------------------------|------------------------------------|------|
| Income per capita | ISTAT | 2018 |
| Inequality | MEF | 2018 |
| Deprivation | ISTAT | 2016 |
| Fiscal capacity | MEF | 2018 |
| Telework | Authors calculation (ISTAT & OECD) | 2020 |
| Intensive care units | ISTAT | 2013 |
| Extra Mortality | ISTAT | 2020 |
| Infected Individuals | Italian Civic Protection | 2020 |
| Cumulative Infected Individuals | Italian Civic Protection | 2020 |

population density and economic poverty. More specifically, indicators for each dimension are computed and then transformed in percentage deviation from the national mean, and finally aggregated together with equal weights.

In addition, we exploit the *Telework* index to understand the extent to which the possibility to work from remote positions might have influenced the message conveyed by individuals, given the fear to lose their job during the pandemic. In particular, this indicator is computed as the linear combination of the product between the telework coefficient by sector, as estimated in a OECD study [15], and the share of revenues generated in the considered province in the correspondent sector, with respect to the overall revenues generated in the underlying administrative unit, provided by the Italian National Institute of Statistics (ISTAT).

However, users opinions might be influenced not only by pre-existing socio-economic conditions, but also by the intensity with which the COVID-19 hit Italian provinces. In particular, the evolution of the pandemic might be a relevant determinant of the sentiment expressed. For this reason we take into account the *extra mortality rate* computed by ISTAT through a comparison of the mortality rate of 2020 with respect to the mean mortality of the previous 5 years. Furthermore, we consider the percentage of infected individuals (*Infected Individuals (%)*), representing the number of confirmed cases in a specific province with respect to the overall local population.

As a matter of fact, online feelings might be dependent not only on the current level of intensity of the pandemic, as the severity of COVID-19 of previous months might induce long lasting emotions in Twitter users. Therefore, we consider as an additional regressor the *Cumulative Infected Individuals (%)*, computed as the ratio between the overall number of infected individuals since the beginning of the pandemic in a given province and the correspondent number of residents.

Finally, we plug in the model the number of intensive/sub-intensive care units (*Intensive care units*), since they constitute a good proxy of the capacity of local healthcare infrastructure to provide adequate medical cares to patients with serious respiratory diseases. In Table 2, we show the descriptive statistics of the time fixed socio-economic variables and of the number of intensive care units. In Tables 3 and 4

Table 2 Descriptive statistics of the time fixed variables

| | Min | Median | Max | Mean | Sd |
|----------------------|---------|---------|---------|---------|---------|
| Income per capita | 13,475 | 18,234 | 25,674 | 18,043 | 2,864 |
| Inequality | 0.563 | 0.795 | 0.910 | 0.780 | 0.064 |
| Deprivation | -85.236 | 0.223 | 320.276 | 38.413 | 89.830 |
| Fiscal capacity | 240.732 | 466.926 | 930.677 | 487.311 | 155.019 |
| Telework | 0.213 | 0.236 | 0.316 | 0.237 | 0.015 |
| Intensive care units | 4 | 50 | 755 | 72.095 | 94.959 |

Table 3 Descriptive statistics for time-varying epidemiological variables during the first wave of contagion

| | Min | Median | Max | Mean | Sd |
|----------------------|--------|--------|-------|-------|-------|
| Extra mortality | -0.157 | 0.201 | 3.631 | 0.483 | 0.682 |
| Infected individuals | 0.0001 | 0.001 | 0.007 | 0.001 | 0.001 |

Table 4 Descriptive statistics for time-varying epidemiological variables during the second wave of contagion

| | Min | Median | Max | Mean | Sd |
|---------------------------------|--------|--------|-------|-------|-------|
| Extra mortality | -0.096 | 0.194 | 0.544 | 0.203 | 0.121 |
| Infected individuals | 0.007 | 0.026 | 0.062 | 0.026 | 0.011 |
| Cumulative infected individuals | 0.008 | 0.030 | 0.070 | 0.031 | 0.013 |

we present the descriptive statistics of the epidemiological time-varying regressors for the first and second wave of contagion. Since the first confirmed case of COVID-19 in Italy was officially registered on the 21st of February 2020 and our first analysed period covers the period 2nd March–26th April, we exclude the cumulative number of infected individuals variable from the analysis of the first wave of contagion, as it would almost coincide with the amount of infected individuals, leading to problems of collinearity in the estimation of the regression models.

4 Methodology

4.1 *Extracting the Online Sentiment*

Sentiment analysis enables the identification of subjective information in written or spoken text. Using a specific lexicon for every language, it is possible to provide a polarity value that carries the positive, negative or neutral connotation of each term. Two main approaches are described in literature to perform sentiment analysis: machine learning approach and lexicon based approach. Machine learning approaches use sample data, known as training data, in order to develop mathematical models able to make prediction or classification. In the field of text mining, the major drawback of these approaches is the fact that performances strongly

depend on the quality and on the domain of trained data. The second class of approaches, i.e. lexicon based approaches, uses opinion words that denote negative or positive opinions building a lexicon. These approaches require the development of a lexicon to be used for the classification phase. As the largest part of developed lexicons focuses on the English language, Italian lexical databases are often created by translating and adapting the English ones. However, the translation of a large number of Italian tweets to English language can be expensive and results tend to be less accurate.

In this work we leverage SentITA, an Italian lexicon-based approach to compute the sentiment of tweets [24]. The tool relies on a combination of two datasets, which amount to 15,000 positively and negatively labelled sentences, and then combined with 90,000 Wikipedia neutral sentences to train the model properly. Finally, the overall dataset was used to train a deep learning model, namely an Attentional Bidirectional Recurrent Neural Network with LSTM cells, that operates at word level. For each tweet, SentITA provides two polarity signals ranging from 0 to 1, one for positive sentiment and one for negative sentiment, as explained also in other studies [24, 30].

We used the model to extract polarity values from each geo-located tweet, and then we computed the average positive (negative) sentiment for each province in a given period of time by computing the mean positive (negative) signal of tweets originated from a given province and published in the period of interest.

We further investigated the presence of social bots in our data, and their potential influence on the net sentiment expressed in each geographical region. We employed a popular and accurate social bot detection algorithm named BotometerLite [35], and we computed for each geolocated tweet a score which ranges between 0 (human) and 1 (bot). We found that 22.57% of the overall number of geolocated tweets was shared by users with bot score higher than 0.5. However, this percentage progressively reduces for higher thresholds and almost vanishes for scores higher than 0.8 and 0.9, where respectively only 3.27% and 0.36% of tweets were shared by potential bots. We computed the correlation between (a) the sentiment variable computed with the full dataset and (b) the sentiment variable computed after excluding tweets shared by accounts with a bot-score larger than K , for $K \in [0.5, 0.6, 0.7, 0.8, 0.9]$ These correlations range between 0.949 and 0.998 and are always statistically significant (p-val $\simeq 0$), providing evidence that the potential presence of social bots did not affect the measurement of our dependent variable.

4.2 Regression Model

In a first step we investigate the main factors affecting the sentiment of Italian provinces respectively during the first and second wave of contagion. We do this through the application of the following specification equation:

$$Y_{i,t} = \beta_0 + \gamma X_i + \delta Z_{i,t} + u_i + \epsilon_{i,t} \quad (1)$$

Table 5 Descriptive statistics for the online sentiment during the first and second wave of contagion

| | Min | Median | Max | Mean | Std |
|-------------------------|--------|--------|-------|--------|-------|
| First period sentiment | -0.285 | 0.058 | 0.305 | 0.029 | 0.146 |
| Second period sentiment | -0.208 | -0.050 | 0.205 | -0.039 | 0.065 |

where index i refers to the underlying administrative unit and subscript t indicates the considered unit of time. More specifically, for the first wave we focus on the time frame 2nd March–26th April with 4 non overlapping windows of 14 days, while for the second wave we consider the period 5th October–27th December with 6 non overlapping windows of 14 days.²

Since in the literature there are few other references for the impact of socio-economic and epidemiological variables on the online sentiment of Twitter users in Italy [30], we apply a set of different panel models to assess the robustness of our results. In particular, we estimate Random and Fixed effects model, and we use the Hausman test to identify the preferred model. However, the idiosyncratic component of the error term $\epsilon_{i,t}$ might be correlated with some of the explanatory variables, leading to the issue of endogenous regressors and biased results. To deal with this problem, we rely on a Generalized Methods of Moments (GMM) estimator [2]. We limit the number of instruments to two per each time-varying regressor, exploiting lags 2 and 3, to guarantee a parsimonious usage [28]. The reason for this is that adopting too many instruments, reduce the power properties of the Hansen test [9] and lead to a downward-bias standard error [34].

Moreover, in order to verify whether the results are mainly driven by the temporal perspective or instead they are stable across the analysed period, we perform a cross section analysis based on Ordinary Least Squares (OLS).

Across all the estimated models, the dependent variable represents the average sentiment at province level, computed as the mean net sentiment of all tweets geolocalized in a certain province in the period of reference. In particular, as the SentIta library associates to each tweet a positive and negative score, the net sentiment score is computed as the difference between the positive and the negative value. In Table 5 we show the descriptive statistics for the online sentiment during the first and second wave of contagion (see Fig. 3 for the geographical distribution of the online sentiment over the two analysed periods).

For what concerns the right-hand side of the equation, X_i is a matrix of time fixed regressors covering socio-economic, health infrastructure capacities and telework feasibility dimensions described in Sect. 3.

By plugging the set of *socio-economic* variables (plus the number of intensive/sub-intensive care units) we control for the fact that the heterogeneity

² The term u_i is the time-fixed error component, while the term $\epsilon_{i,t}$ is the time-varying error component. β_0 is the intercept of the model and γ and δ are vectors of parameters referring to socio-economic and epidemiological variables and need to be estimated.

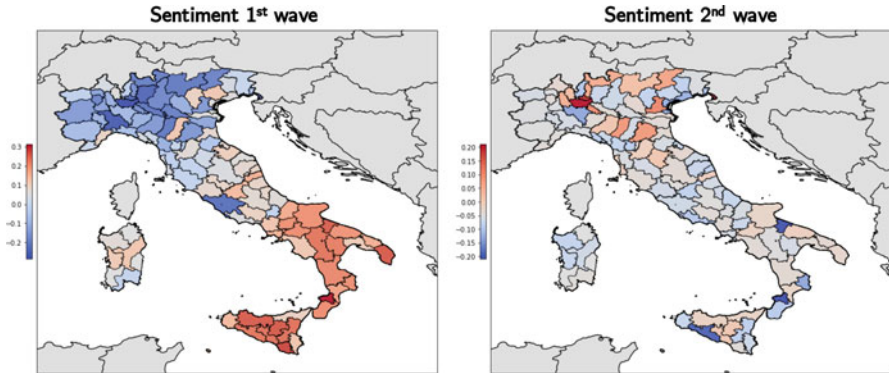


Fig. 3 Geographical distribution of the online sentiment on Twitter during the two periods of interest, computed at province level, which is used as dependent variable in our regression model. The colorbar is centered at the median value of the distribution

in the sentiment across different provinces might be driven by wealth, economic disparities and capacities of the local health infrastructure. Indeed, as the restrictive measures adopted during the lockdown severely hit several production sectors, value chains, trade exchange and the sanitary system risked to be overburden by the number of patients, we suspect that these variables might be relevant determinants to explain the sentiment of Twitter users.

Moreover, as the COVID-19 pandemic was primarily a healthcare emergency, we include into the model the matrix $Z_{i,t}$, which encompasses time variant regressors addressing the *epidemiological* dimension.

In a second step, we study the main factors driving the change in the feelings expressed by Twitter users between the first and second wave of contagion, through the application of the following OLS regression:

$$\bar{Y}_{i,2} - \bar{Y}_{i,1} = \beta_0 + \gamma X_i + \delta(Z_{i,2} - Z_{i,1}) + u_i + \epsilon_{i,t} \tag{2}$$

In this case the dependent variable is represented by the difference between the average sentiment respectively during the second and first wave of contagion in each province, while the term $(Z_{i,2} - Z_{i,1})$ represents the difference between the time varying variables as measured in the two analysed periods. For both the dependent variable and the time varying regressors we rely on an absolute difference of the correspondent factors between the second and first wave of contagion, so that positive values corresponds to positive differences and negative values to negative differences.³

³ On the other hand, using a percentage variation might induce larger complexity in interpreting the sign of the variables since changes in the sign of variables might lead to positive or negative values of the indicators, independently on the fact the factor is growing or diminishing. Moreover, these variables are not affected by size factors, therefore, using percentage variation might risk to

5 Results and Discussion

5.1 *The First Wave of Contagion*

In this section we show the results of our regression models, with a focus on the first wave of contagion. In a first step, we estimate a set of different panel and cross section models to assess the robustness of our results (see Table 6). We rely on the formal Hausman test to distinguish between a Random and a Fixed effects model. The p-value larger than 0.05 allows to accept the null hypothesis and is clearly in favour of the former specification. As a consequence, we apply a Random effects panel model⁴ based on White heteroskedasticity robust standard errors [33], since the Breusch Pagan test clearly rejects the hypothesis of constant variance across the error terms.

This model suggests that during the first wave of contagion epidemiological variables have a strong impact on online feelings. Indeed, both the extra mortality rate and fraction of infected individuals have negative and statistically significant coefficients. This pattern reveals that the administrative units hit by the virus with stronger intensity in terms of deaths and number of contagions tend to convey more pessimistic messages.

Moreover, we observe a negative relationship between the number of intensive and sub-intensive care units at hospitals and the online sentiment. This can be explained by the fact that those areas with a higher capacity in the healthcare infrastructures also experienced an overburden due to a more severe impact of the COVID-19 pandemic, thus leading to more pessimistic feelings.

We obtain significant coefficients also for the income per capita and the deprivation index variables. Indeed, individual wealth shows a negatively interplay with Twitter users feelings, revealing that more positive emotions are associated with areas characterized by lower levels of income per capita. This apparently counter-intuitive result can be explained by the strong correlation between the intensity of the pandemic and the income level of Italian areas. This means that the areas which were more strongly affected by the pandemic were rich territories. Indeed, focusing on the first wave and comparing the variable *income per capita* with *Infected Individuals* and then with *Extra Mortality*, we find respectively correlation coefficients equal to 0.639 (p-val \simeq 0) and 0.547 (p-val \simeq 0). Consequently, this suggests that during the first wave, the Twitter sentiment was driven by the intensity of the pandemic, as the sentiment was significantly lower in areas subject to higher contagions and extra mortality rates. This is confirmed by the negative coefficient for

overestimate even small variations in provinces having a low value in the first wave of contagion for the considered variable.

⁴ For completeness, we report also the model estimate according to a Fixed effects model, that is unbiased (as the Random effect, due to the result of the Hausman Test), but less efficient. The results for the time-varying covariates in the Fixed effects confirms the results of the Random effect.

Table 6 This table shows the results of the estimation of a OLS, Random effect, Fixed effects and GMM models, with respect to the first wave of contagion period

| | <i>Dependent variable:</i> | | | |
|----------------------------|----------------------------|-----------------------|--------------------|---------------------|
| | Online Sentiment | | | |
| | (OLS) | (RE) | (FE) | (GMM) |
| Income pc | −0.407*** (0.123) | −0.262** (0.128) | | −0.272* (0.160) |
| Fiscal Capacity | −0.116 (0.084) | −0.024 (0.081) | | −0.012 (0.086) |
| Deprivation | 0.186* (0.103) | 0.245* (0.128) | | 0.351*** (0.113) |
| Inequality | 0.024 (0.074) | −0.059 (0.095) | | −0.040 (0.092) |
| Telework | 0.035 (0.083) | 0.030 (0.104) | | −0.020 (0.112) |
| Intensive care units | −0.185*** (0.066) | −0.143* (0.085) | | −0.091 (0.090) |
| Extra Mortality | −0.233** (0.111) | −0.221** (0.104) | −0.147* (0.083) | −0.117* (0.071) |
| Infected Individuals | −0.163* (0.096) | −0.105* (0.059) | −0.122 (0.119) | −0.249** (0.176) |
| Observations | 105 | 420 | 420 | 420 |
| Adjusted R ² | 0.775 | 0.276 | 0.218 | 0.308 |
| Breusch-Pagan test | 0.021 | 1.12*10 ^{−5} | | |
| Hausman Test | | 0.228 | | |
| Hansen Test | | | | 0.635 |
| Arellano Bond test order 1 | | | | 0.003 |
| Arellano Bond test order 2 | | | | 0.395 |

Note: *p<0.1; **p<0.05; ***p<0.01

wealth due to the strong correlation of this variable with pandemic intensity, which leads territories experiencing the worst online feelings to be also the richer territories of Italy. Similarly, more optimistic feelings are expressed in areas with higher socio-economic imbalance, as suggested by the positive coefficient for the deprivation index. Finally, inequality, fiscal capacity and telework seem not to significantly affect Twitter users sentiments.

In a second step, we verify the consistency of our previous results, through the application of a GMM model, where we use the second and third lag of the time-varying regressors as instrumental variables. In particular, we apply this model, since the error term might be correlated with some of the explanatory variables, thus inducing biased estimates. Overall, the GMM model show similar results with respect to the Random effect model and the only exception is represented by

the Intensive care units regressor, that show is still negative, but not statistically significant.⁵

Finally, we assess whether the results were mainly driven by the temporal perspective or instead they were stable across the analysed period, through a cross section analysis based on Ordinary Least Squares (OLS). The results corroborate our previous estimates since administrative units with more optimistic sentiments are those with lower income per capita and higher deprivation. Moreover, we confirm that areas more severely hit by the COVID-19 in terms of extra mortality and number of infected individuals experience the more pessimistic sentiments.

5.2 The Second Wave of Contagion and Main Differences with the Previous Period

In this section, we repeat the analysis performed in Sect. 5.1, to analyse the main determinants of online sentiment during the second wave of contagion. Results are shown in Table 7. The estimated models provide coherent results and display evidence of a positive and statistically significant coefficient for the income per capita and the deprivation index, meaning that areas with larger wealth and socio-economic imbalance expressed the more optimistic messages over the time frame October–December 2020. Moreover, we identify a positive interplay between the telework coefficient and the dependent variable, suggesting that the possibility to work from remote positions during the pandemic contributed to reduce negative feelings among Twitter users. This might reveal how the possibility to continue to perform jobs activities even during the healthcare emergency, contributed to reduce the perception of the negative effects induced by the pandemic, while the sentiments were significantly more pessimistic in areas where this approach was less feasible.

In addition, we highlight the low relevance of epidemiological variables during the second wave of contagion. Indeed, the amount of infected individuals, the extra mortality rate and the number of Intensive care units do not appear as statistically significant factors affecting the online sentiment.⁶ Moreover, the cumulative number of infected individuals variable has a positive coefficient, meaning that provinces

⁵ We perform a Hansen test to verify that the identified instruments are valid and robust. A p-value = 0.635 is in favour of the null hypothesis, which states the exogeneity of the instruments. Moreover, the GMM estimator provides reliable results in case the first differenced error term does not display second order autocorrelation and for this reason, we perform the Arellano-Bond test. In particular, we show evidence of first order autocorrelation, but absence of statistically significant autocorrelation of order 2 (AR(1) p-value = 0.003, AR(2) p-value = 0.395), contributing to the validity of the results.

⁶ The only exception is represented by the OLS model where the extra mortality rate has a positive and significant coefficient. This suggest the disconnection between epidemiological variables and Twitter users sentiment, since more positive feelings were expressed by provinces with higher number of additional deaths.

Table 7 This table shows the results of the estimation of a OLS, Random effect, Fixed effects and GMM models, with respect to the second wave of contagion period

| | <i>Dependent variable:</i> | | | |
|---------------------------------|----------------------------|-----------------------|---------------------|-------------------|
| | Online Sentiment | | | |
| | (OLS) | (RE) | (FE) | (GMM) |
| Income pc | 0.524** (0.228) | 0.283* (0.170) | | 0.218* (0.123) |
| Fiscal Capacity | 0.012 (0.123) | 0.017 (0.122) | | -0.043 (0.103) |
| Deprivation | 0.419** (0.196) | 0.253* (0.151) | | 0.221* (0.115) |
| Inequality | 0.219 (0.143) | 0.172 (0.118) | | 0.157 (0.099) |
| Telework | 0.314* (0.158) | 0.259* (0.149) | | 0.245* (0.134) |
| Intensive care units | -0.143 (0.129) | -0.085 (0.107) | | -0.082 (0.093) |
| Extra Mortality | 0.194* (0.110) | 0.014 (0.054) | 0.062 (0.075) | 0.019 (0.059) |
| Infected Individuals | 0.391 (0.343) | -0.076 (0.056) | -0.030 (0.050) | -0.002 (0.048) |
| Cumulative Infected Individuals | 0.217** (0.105) | 0.124* (0.071) | 0.272*** (0.083) | 0.186* (0.100) |
| Observations | 105 | 630 | 630 | 630 |
| Adjusted R ² | 0.473 | 0.268 | 0.256 | 0.328 |
| Breusch-Pagan test | 0.034 | 2.24*10 ⁻⁸ | | |
| Hausman Test | | 0.241 | | |
| Hansen Test | | | | 0.557 |
| Arellano Bond test order 1 | | | | 0.008 |
| Arellano Bond test order 2 | | | | 0.281 |

Note: *p<0.1; **p<0.05; ***p<0.01

with more positive feelings where those experiencing larger volumes of contagions since the beginning of the pandemic, confirming a misalignment between the evolution of the pandemic and Twitter users feelings. This result is reasonable considering that during the second wave of contagion, Italian regions were experiencing more similar percentages of infected individuals and number of deaths with respect to the first wave. For instance, during the first wave, the distribution of extra-mortality rates ranges between -0.157 and 3.631 (25th percentile = 0.078, Median = 0.201, 75th percentile = 0.630), while during the second wave the distribution of extra-mortality rates shrinks between -0.096 and 0.544 (25th percentile = 0.115, Median = 0.194, 75th percentile = 0.262). Similarly, during the period March-May 2020, the regions of Lombardia and Emilia-Romagna together accounted for 50% of overall national

contagions, and considering also Piemonte and Veneto, the percentage reaches the 70%. On the other hand, in the period October-December 2020, Lombardia and Veneto accounted for only 33% of total contagions and including Piemonte and Veneto they cover the 53% of overall infected individuals. These patterns, clearly suggest that during the first wave of contagion there was a strong difference in terms of pandemic intensity especially between Northern and Southern areas which strongly affect the online sentiment. Conversely, the COVID-19 pandemic is much more uniformly widespread across Italian territories during the second wave and differences in terms of contagions and extra mortality rates do not seem to have an impact on the online feelings of Twitter users. In addition, the first wave of contagion was more severe in terms of deaths and stress of hospitals and healthcare infrastructures with respect to the second [5, 7, 8, 23]. In fact, although the number of spotted infected individuals was significantly higher during the second period (due to the significantly higher capacity to execute swabs in Italian healthcare infrastructures), the extra-mortality rates were higher during the first wave. In addition, due both to the better knowledge of the COVID-19 disease and to the increased capacity of hospitals, during the second wave many individuals were efficiently and effectively cured without requiring intensive and sub-intensive therapies, reducing their saturation rates w.r.t. to the period March-May 2020 [27]. During the second wave of contagion, all these factors contributed to reduce both the perception of the severity of the COVID-19 pandemic and relevance of healthcare variables in driving the Twitter users' sentiment w.r.t. the first wave of contagion.

Overall, we find evidence of competing behaviours in the determinants of the online sentiments over the two analysed periods. Indeed, in the first wave of contagion, the sentiments were driven by the local intensity of the pandemic, which was mainly perceived as an healthcare emergency. On the other hand, in the period October-December 2020 a larger relevance to explain Twitter users emotions is accounted by socio-economic variables, suggesting how in a second step it might have prevailed the awareness of the financial crisis triggered by the pandemic.

Due to these strong differences in the sentiments among the two analysed periods, we further investigate the drivers of the change in the province feelings expressed between the first and second wave of contagion. To do this we apply an OLS regression, according to Eq. (2) (see Table 8). We show that the main determinants of this variation in the conveyed emotions are represented by the socio-economic variables. Indeed, we identify that areas experiencing the strongest improvement in the sentiments communicated, are those provinces with higher income per capita. Moreover, a relevant driver is represented by the telework index, meaning that a larger possibility to work from remote, contributed to significantly improve the expressed feelings. Moreover, the change in the sentiment was positively affected by the number of Intensive care units, revealing that in the long term the higher capabilities of healthcare infrastructures contributed to perceive better emotions. This is coherent with the fact that the more positive variations in the feelings conveyed was experienced in the North of Italy.

Table 8 This table shows the results of the estimation of an OLS regression for the estimation of Eq. (2)

| | <i>Dependent variable:</i> |
|--|-----------------------------|
| | Online sentiment difference |
| Income pc | 0.498*** (0.153) |
| Fiscal Capacity | 0.096 (0.083) |
| Deprivation | -0.020 (0.124) |
| Inequality | 0.066 (0.092) |
| Telework | 0.197* (0.105) |
| Intensive care units | 0.460** (0.187) |
| Extra Mortality Difference | -0.237 (0.202) |
| Infected Individuals Difference | 0.366 (0.651) |
| Cumulative Infected Individuals Difference | -0.353 (0.680) |
| Observations | 105 |
| Adjusted R ² | 0.700 |
| Breusch-Pagan test | 0.012 |

Note: *p<0.1; **p<0.05; ***p<0.01

6 Conclusion

In this paper we analyse the online sentiment conveyed by Italian Twitter users during COVID-19 pandemic leveraging a large-scale dataset encompassing over 2.5 M tweets. Focusing on a subset of 660 K geo-located tweets shared during the first and second wave of contagion, we investigate the main factors affecting the online feelings among socio-economic and epidemiological variables through the application of panel and cross-section regression models.

Overall, we observe a strong heterogeneity between the two periods of observation. In particular, we find that during the first wave of contagion sentiments are strongly affected by the severity of the pandemic. Indeed, more pessimistic sentiments are expressed by administrative units subject to a higher number of contagions and extra mortality rate. Moreover, we find evidence that more positive feelings are conveyed by areas with lower income per capita and larger deprivation.

On the other hand, we identify that during the second wave of contagion, Twitter users emotions are mainly affected by socio-economic variables, as more optimistic

sentiments are communicated in provinces with larger wealth and telework feasibility. In addition, we highlight a misalignment between the feelings expressed and the intensity of the pandemic, since more positive messages are conveyed by areas characterized by an higher cumulative amount of infected individuals.

Finally, we observe that during the first period there is a sharp geographical diversity in the sentiment, with Southern areas communicating the more optimistic messages. On the other hand, during the second wave of contagion, the feelings are more homogeneous across areas, without a clear separation between the North and South of Italy. Overall, we observe that the stronger improvements in the emotions are experienced by Northern provinces, which are characterized by larger wealth, telework index and number of intensive care units.

This paper sheds light on the factors affecting the evolution of individuals sentiments. In particular, we observe that during the first wave of contagion, the COVID-19 pandemic is perceived as a healthcare emergency and the epidemiological variables account for higher relevance in explaining the feelings. On the other hand, during the second period, the socio-economic variables play a more significant role in the influence of the online sentiment. Particularly interesting is the significance of the telework index, which suggests that individuals have progressively attributed higher importance to the possibility to work remotely.

From a methodological point of view, we contribute to the analysis of potential drivers of the online sentiment, providing relevant insights for further research. Indeed, understanding the importance of socio-economic variables during pandemics is still at an embryonic stage. In addition, this work could provide significant clues to policy makers on the impact of the pandemic on citizens' sentiments. These can be exploited in order to design targeted and effective communication preventing the heterogeneous spread of feelings across Italy.

Overall, our research presents some limitations. First, although we rely on a large dataset of tweets over the two periods, one could further extend the amount of online messages in order to increase the representatives of the sample and consolidate the findings. Second, the usage of Twitter is more widespread among specific population cohorts, therefore the sample considered for the analysis might not be representative of the entire population and results might not be generalized. Third, we study the whole set of collected tweets, without performing more detailed analyses focused on specific sub-topics associated to the COVID-19 pandemic. In this direction, as potential future research extension, we plan to perform a comparative analysis of online sentiment across multiple countries, focusing on the specific topic of the vaccination campaign that has recently kicked off on global scale.

References

1. Aiello LM, Quercia D, Zhou K, Constantinides M, Šćepanović S, Joglekar S (2020) How epidemic psychology works on social media: Evolution of responses to the covid-19 pandemic. arXiv preprint arXiv:2007.13169

2. Arellano M, Bond S (1991) Some tests of specification for panel data: Monte carlo evidence and an application to employment equations. *Rev Econ Stud* 58(2):277–297
3. Basile V, Caselli T (2020) 40twita 1.0: An collection of italian tweets during the covid-19 pandemic. <http://twita.di.unito.it/dataset/40wita>
4. Bento AI, Nguyen T, Wing C, Lozano-Rojas F, Ahn Y-Y, Simon K (2020) Evidence from internet search data shows information-seeking responses to news of local covid-19 cases. *Proc Natl Acad Sci* 117(21):11220–11222
5. Berardi C, Antonini M, Genie MG, Cotugno G, Lanteri A, Melia A, Paolucci F (2020) The covid-19 pandemic in Italy: policy and technology impact on health and non-health outcomes. *Health Policy and Technology* 9(4):454–487
6. Bonaccorsi G, Pierri F, Cinelli M, Porcelli F, Galeazzi A, Flori A, Schmidh AL, Valensise CM, Scala A, Quattrociochi W, Pammolli F (2020) Economic and social consequences of human mobility restrictions under covid-19. *Proc Natl Acad Sci* 117(27):15530–15535
7. Bontempi E (2021) The Europe second wave of covid-19 infection and the Italy “strange” situation. *Environ Res* 193:110476
8. Borghesi A, Golemi S, Carapella N, Zigliani A, Farina D, Maroldi R (2020) Lombardy, Italy: Covid-19 second wave less severe than the first? a preliminary investigation
9. Bowsher CG (2002) On testing overidentifying restrictions in dynamic panel data models. *Econ Lett* 77(2):211–220
10. Buckee CO, Balsari S, Chan J, Crosas M, Dominici F, Gasser U, Grad YH, Grenfell B, Halloran ME, Kraemer MUG, Lipsitch M, Metcalf CJE, Meyers LA, Perkins TA, Santillana M, Scarpino SV, Viboud C, Wesolowski A, Schroeder A (2020) Aggregated mobility data could help fight covid-19. *Science* 368(6487):145–146
11. Bwire G, Munier A, Ouedraogo I, Heyerdahl L, Komakech H, Kagirita A, Wood R, Mhlanga R, Njanpop-Lafourcade B, Malimbo M, Makumbi I, Wandawa J, Gessner BD, Orach CG, Mengel MA (2017) Epidemiology of cholera outbreaks and socio-economic characteristics of the communities in the fishing villages of Uganda: 2011–2015. *PLoS Negl Trop Dis* 11(3):1–19, 03
12. Chew C, Eysenbach G (2010) Pandemics in the age of Twitter: Content analysis of tweets during the 2009 H1N1 outbreak. *PLoS One* 5(11):1–13
13. Chung W (2015) emood: Modeling emotion for social media analytics on Ebola disease outbreak. In: *International Conference of Information Systems*
14. Culotta A (2010) Towards detecting influenza epidemics by analyzing twitter messages. In: *Proceedings of the First Workshop on Social Media Analytics, SOMA '10*. Association for Computing Machinery, New York, pp 115–122
15. Espinoza R, Reznikova L (2020) Who can log in? The importance of skills for the feasibility of teleworking arrangements across OECD countries. In: *OECD Social, Employment and Migration Working Papers*
16. Fung IC-H, Tse ZTH, Cheung C-N, Miu AS, Fu K-W (2014) Ebola and the social media. *The Lancet* 384(9961):2207
17. Guarino S, Pierri F, Di Giovanni M, Celestini A (2021) Information disorders during the covid-19 infodemic: The case of Italian Facebook. *Online Social Networks and Media* 22:100124
18. Jain VK, Kumar S (2015) An effective approach to track levels of influenza-a (h1n1) pandemic in India using Twitter. *Procedia Computer Science* 70:801–807. *Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems*
19. Jordan SE, Hovet SE, Fung ICH, Liang H, Fu KW, Tse ZTH (2019) Using twitter for public health surveillance from monitoring and prediction to public response. *Data* 4(1):1–20
20. Kaila A, Prasad RP (2020) Informational Flow on Twitter—Corona Virus Outbreak—Topic. *International Journal of Advanced Research in Engineering and Technology (IJARET)* 11(3):128–134
21. McInnes CJ, Hornmoen H (2018) ‘Add twitter and stir’: The use of twitter by public authorities in Norway and UK during the 2014-15 Ebola outbreak. *Observatorio* 12(2):23–46
22. Medford RJ, Saleh SN, Sumarsono A, Perl TM, Lehmann CU (2020) An “Infodemic”: Leveraging high-volume Twitter data to understand public sentiment for the COVID-19

- outbreak. In: *Open forum infectious diseases*, vol. 7(7). Oxford University Press, New York, p ofaa258. medRxiv
23. Olivieri A, Palù G, Sebastiani G (2021) Covid-19 cumulative incidence, intensive care, and mortality in Italian regions compared to selected European countries. *Int J Infect Dis* 102:363–368
 24. Pelosi S (2015) Sentita and doxa: Italian databases and tools for sentiment analysis purposes. In: *Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it*, pp 226–231
 25. Pierri F, Perry BL, DeVerna MR, Yang K-C, Flammini A, Menczer F, Bryden J (2022) Online misinformation is linked to early COVID-19 vaccination hesitancy and refusal. *Scientific reports* 12(1):1–7
 26. Rahman MM, Ali GGMN, Li XJ, Paul KC, Chong PHJ (2020) Twitter and census data analytics to explore socioeconomic factors for post-covid-19 reopening sentiment. medRxiv
 27. Richardson E, Aissat D, Williams GA, Fahy N, et al (2020) Keeping what works: remote consultations during the covid-19 pandemic. *Eurohealth* 26(2):73–76
 28. Roodman D (2009) How to do xtabond2: An introduction to difference and system gmm in stata. *Stata J* 9(1):86–136
 29. Samuel J, Ali GGN, Rahman MM, Esawi E, Samuel Y (2020) COVID-19 public sentiment insights and machine learning for tweets classification. medRxiv 4:1–21
 30. Scotti F, Magnanimi D, Urbano VM, Pierri F (2020) Online feelings and sentiments across Italy during pandemic: investigating the influence of socio-economic and epidemiological variables. In: *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, New York, pp 453–459
 31. Spelta A, Flori A, Pierri F, Bonaccorsi G, Pammolli F (2020) After the lockdown: simulating mobility, public health and economic recovery scenarios. *Sci Rep* 10(1):1–13
 32. Towers S, Afzal S, Bernal G, Bliss N, Brown S, Espinoza B, Jackson J, Judson-Garcia J, Khan M, Lin M, Mamada R, Moreno VM, Nazari F, Okuneye K, Ross ML, Rodriguez C, Medlock J, Ebert D, Castillo-Chavez C (2015) Mass media and the contagion of fear: The case of Ebola in America. *PLoS One* 10(6):1–13, 06
 33. White H (1980) A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. In: *Econometrica: Journal of the Econometric Society*, pp 817–838
 34. Windmeijer F (2005) A finite sample correction for the variance of linear efficient two-step gmm estimators. *J Econ* 126(1):25–51
 35. Yang K-C, Varol O, Hui P-M, Menczer F (2020) Scalable and generalizable social bot detection through data selection. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol 34, pp 1096–1103
 36. Yang K-C, Pierri F, Hui P-M, Axelrod D, Torres-Lugo C, Bryden J, Menczer F (2021) The covid-19 infodemic: Twitter versus Facebook. *Big Data Soc.* 8(1):20539517211013861
 37. Zarrad A, Jaloud A, Alsmadi I (2014) The evaluation of the public opinion—a case study: Mers-cov infection virus in ksa. In: *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC '14*. IEEE Computer Society, New York, pp 664–670

Inferring Degree of Localization and Popularity of Twitter Topics and Persons Using Temporal Features



Aleksey Panasyuk, Kishan G. Mehrotra, Edmund Szu-Li Yu,
and Chilukuri K. Mohan

Abstract Useful information can be extracted by analyzing the temporal distributions of both social media user account creation and message traffic data. When applied over message traffic, the approach can differentiate top trending topics and persons in different geographical regions. Our analysis can help discover whether (and where) an influencer's followers are localized, even in the absence of geospatial tags. An important application is in finding local experts in a social network, by identifying which experts are relevant to the geographic region of interest. We demonstrate how several temporal features can be utilized for distinguishing local vs. global influencers. For global influencers, spatiotemporal analysis helps understand the evolution of their popularity over time. We can also infer the number of followers that were gained in a specified period, which assists in estimating link creation times. Thus, temporal features can assist in deducing and utilizing information about the numbers and locations of influencers' followers.

Keywords Global influence · Local expert · Geo-influencer · Account creation time distribution · Inferring link creation times · Temporal features · Social networks · Network dynamics

A. Panasyuk

Information Fusion Technology Branch, Air Force Research Lab, Rome, NY, USA

Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, USA

e-mail: apanasyu@syr.edu

K. G. Mehrotra · E. S.-L. Yu (✉) · C. K. Mohan

Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, USA

e-mail: mehrotra@syr.edu; esyu@syr.edu; ckmohan@syr.edu

1 Introduction

Identifying authoritative users (experts) or influencers on social networks is an important topic of research.¹ Local expert finding is important for many applications such as answering local information needs. The area of influence and expertise may be localized within a small geographical area for some influencers, and be much broader for others.

The problem with local expert finding in social networks and on Twitter, in particular, is the lack of geo-information. Less than 1% of message traffic contains geo-information, and available information about a user's location is limited to a self-reported textual field that may not be filled in. In this research, we illustrate an alternative approach for how the creation times can be used to infer geo-information.

On Twitter, every user and every message has a creation timestamp. For a group of users or a group of messages, the creation times can be used to help determine whether the group is concentrated in a single time zone or is spread out more globally. The Coordinated Universal Time (UTC) offset² can be identified for a group that is from a specific time zone. For a global group (such as the followers of a global influencer), the daily changes in followers can be inferred and used for studying the influencer's evolving popularity.

This work extends our previous paper [31], the time-based features discussed have applications related to (i) local expert finding in social networks, (ii) inferring when followers joined an influencer, and (iii) understanding popular trending topics from message traffic relevant to a specific geographic area. Main portions of the code are hosted on GitHub.³

The methods in this paper maintain user's privacy because the location inference is at the timezone level. Care was taken, during data collection, to minimize impact from bots; software programs that send out automated posts on Twitter. For analysis over influencer's followers, we focused on influencers that have been verified by Twitter to be legitimate, but in general followers-friends ratio, tweet frequency, number of times added to favorites, and other features such as screen name length can be used for identifying real influencers [1]. When analyzing message traffic, to ensure equal representation for each user, it is recommended to focus on a single message per user.

The rest of the paper is structured as follows. Section 2 reviews prior research related to local expert finding. Section 3 shows how group creation times can be used in a time distribution and how this distribution can be used for predicting the UTC offset. Section 4 analyzes the temporal distribution of message traffic.

¹ We use the term influencer and expert interchangeably when referring to an authoritative user.

² UTC is the time standard used globally, defined by the International Telecommunication Union Recommendation (ITU-R TF.460-6); it is a refinement of previous time standards such as Greenwich Mean Time. For instance, the UTC offset is -5 for the time zone that includes the northeastern USA.

³ <https://github.com/apanasyu>.

Section 5 analyzes the variations in the number of followers and illustrates how those can be used for understanding daily followers gained. This is useful for link inference and understanding evolving popularity of global influencers. Section 6 describes a classifier for the discrimination of local vs. global influencers. Finally, Sect. 7 presents our conclusions and future research directions.

2 Related Research

The problem of finding authoritative users is known as expert finding; this is a well-studied problem with research going back over a decade, and has gained popularity within the information retrieval community since it was included in the TREC enterprise track [2]. A recent survey by Husain et al. [3] reports that a majority of the expert finding systems were used in: (i) the academic domain (research collaborations), (ii) enterprise (experts for offering formal help related to development), (iii) medicine (medical experts), (iv) online knowledge sharing communities, (v) online forums, and (vi) social media (finding experts from various social networks like Twitter and Facebook).

Expert finding methods assume that individuals' published documents are relevant to their expertise with different degrees of a match, and they focus on modeling the associations between these documents and candidate experts.

Our research deals with social networks. Lappas et al. [4] give an early survey on expert finding in social networks, which typically involves (i) using text content posted by expert candidates and (ii) using the expert candidates' online social connections. Two best-known algorithms that exploit link structure to find authorities are based on PageRank [5] and Hyperlink-Induced Topic Search (HITS) [6].

Weng et al. [7] proposed TwitterRank which employs the Latent Dirichlet Allocation (LDA) model to detect the topics of individuals based on their tweets. Then, for each topic, it builds a weighted graph based on the topical similarity between two users and then employs a PageRank algorithm to find topic-specific influential users.

Romero et al. [8] designed an algorithm similar to HITS named Influence Passivity algorithm to quantify the influence of users in a Twitter network. This algorithm utilizes both the structural properties of the network as well as the diffusion behavior among users. Pal et al. [9] proposed an attribute-based approach for identifying experts and potential experts in community question answering. Fifteen features were extracted from the Twitter graph and tweets posted by the users, to estimate their levels of expertise on various topics. Clustering (based on the Gaussian mixture model) was used to determine experts, maximizing the likelihood of the data given a number of Gaussian components.

Ghosh et al. [10] proposed a system called Cognos, which represents each user by the metadata of Twitter lists that contain the user, then ranks users based on the similarity score between each user and a topical query. Cognos tends to choose users

that are contained in many lists and whose metadata contains the query. The authors show that their system can identify top users for a particular topic better than graph based approaches.

Separately, research efforts have addressed the task of finding local experts with specialized knowledge focused around a particular location. Local experts are important for many applications such as answering local information needs [11].

Li et al. [12] proposed applying points of interest (POI) as a possible categorization of expertise related to a particular geographic location. Example ‘Chinese Restaurants’ in Los Angeles is a POI topic. High-ranking candidates should be able to answer questions about the locations or the category of locations in the topic. The time user reported being at a POI is seen as an important feature in that frequent visits result in greater familiarity with the location in question [13].

Niu et al. [14] introduced a learning-based method to find local experts on Twitter. They defined multiple classes of features that could impact a user’s local expertise, such as tweet content features (e.g. the TF-IDF score of a topic keyword in the candidate’s tweets) and local authority features (e.g. the distance between the candidate and the query location). Authors found it best to retain only the first check-in during a repeated activity (a user posting multiple times about a newly served dish during the same meal is an example of the same venue during which the user remains in an unchanged location and activity).

In papers that attempt to identify topical experts typically the GPS coordinates and place mentions associated with messages are utilized. Inkpen et al. [15] develop a city, province, and country classifier for monitoring places mentioned in Twitter messages. The issue with focusing only on tweets with GPS coordinates or POI information is that they make up a small portion of the Twitter API stream (around 1%) [16, 17]. The message’s author may specify a textual self-reported location (available for about a third of all users [18]), but geocoding is complicated due to abbreviations, misspellings, blank textual field, and use of multiple alphabets [17]. Jurgens et al. [16] reported that using popular gazetteer solutions GeoNames, DBPedia, GeoLite, and Google’s geocoder were able to each geocode under 4% of users using self-reported location [16]. For users whose location cannot be extracted from their message or profile information, the median location of the user’s friends may be used [19].

Wei et al. [20] attempt to identify local influencers across three US cities using several modified PageRank based algorithms. Their network was built using social activity based interactions retweet, reply, and mention present in over five billion tweets (message contents not analyzed). The influencer’s self-reported location was used for filtering out those influencers that are not from the area such as *@YouTube*. However, it was also shown that limiting users within x miles of the location of interest would filter out other important users, that had a strong local connection spanning beyond 100 km.

Multiple surveys have been written related to Twitter user geolocation [16, 17, 21]. Jurgens et al. [16] reimplemented some of the state-of-the-art models, tested and trained them using their own constructed dataset to ensure fairness of comparison, and found significant performance issues. Mourad et al. [21] proposed a guide for a

standardized evaluation of Twitter user geolocation. Analysis of fifteen models and two baselines illustrated that the choice of effectiveness metric can lead to diverging conclusions. Due to the high levels of noise and the data collection restrictions imposed by the Twitter API the user geolocation remains an unsolved research area.

Other features useful for identifying locations are the time zone and UTC offset [22, 23]. Zannettou, et al. [24] used time zone information to understand the audience targeted by tweets from Russian-linked accounts. But due to privacy reasons, Twitter has made these fields inaccessible in 2018.

Twitter does not keep track of any time information other than identifying when a user account was created, and when a user posts on Twitter. Data for link creation times between users and their followers are not stored, although it can be extracted by performing multiple scans of the Twitter network. For example, Kwak et al. [25] collected daily snapshots of the online relationships of 1.2 million Korean-speaking users for 51 days as well as all of their tweets to estimate popularity dynamics.

This research paper proposes new time-based features based on user and message creation times. Account creation times over influencer's followers are used for predicting the time zone's UTC offset and associated geographic area that the followers belong to. When applied over message traffic, the approach can differentiate top trending topics and persons in different geographical regions. The degree of localization ("localness") is an important concept, with ongoing work in formalizing the notion [26]. Our time-based features are successfully applied in a classifier for predicting local vs. global influencers. The resulting classifier can be applied as a post-processing step for verifying that the local expert is indeed local. The new time-based features are not just limited to inferring location, but can also be used for inferring link creation times for studying the evolution of influencer's popularity.

3 UTC Offset Prediction Based on Account Creation

This section describes how the time zone's UTC offset is predicted from a set of creation times. The creation times can come from a set of users or a set of messages. Section 3.1 describes the dataset; the creation times come from a group of users whose self-reported location is in common and where the location's UTC is known. Section 3.2 describes how a time distribution is formed and how it is used to predict the UTC offset. Section 3.3 describes experiments to find the optimal parameter values used in the proposed approach.

3.1 UTC Offset Dataset

Over 377 million user profiles were analyzed and user groups were chosen based on self-reported location in common. All self-reported locations were turned to

Table 1 Five biggest user groups in UTC offset dataset

| Location | Group size | UTC ^L | Country |
|--------------|------------|------------------|---------|
| london | 2065562 | 0.667 | GBR |
| losangelesca | 1768898 | -7.333 | USA |
| newyorkny | 1425330 | -4.333 | USA |
| chicagoil | 1173340 | -5.333 | USA |
| parisfrance | 1026459 | 1.667 | FRA |

lowercase with punctuation and spacing stripped out. Of particular interest are those self-reported locations that match (i) (City, Province) or (ii) (City, Country Name) in English from GeoNames. The city, country pairs are checked to be unique in that there are no other cities within the country with the same city name. The population of all cities considered in is over five thousand. Major well-known city names are included (without the country name) provided the city is unique and has a population of over one million. Each self-reported location had to be used by at least 250 unique users to ensure a large enough sample size.

The resulting dataset, denoted D_{UTC} , consists of 12,271 groups. Table 1 shows the five most popular locations, the number of users making up each group that use the location, and the UTC offset associated with the location, denoted as UTC^L , using Eq. (1).

$$UTC^L(loc) = \frac{1}{3}UTC(tmz(loc)) + \frac{2}{3}DST(tmz(loc)) \quad (1)$$

GeoNames is used to get the location's time zone⁴ via function tmz . UTC and DST functions are used to obtain the UTC offset during standard and daylight saving time, respectively; these are equal in time zones where daylight saving is not observed. Daylight saving time is typically observed for eight months of the year and is thus given a larger weight.

In our dataset, UTC^L takes 42 possible values ranging from -9.9 to 13.53. Therefore, the corresponding UTC offset interval for our dataset is [-10, 14) (UTC offset -12 and -11 exist, but belong to sparsely populated islands and therefore not of interest). Table 1 describes the attributes of the five largest user groups in the dataset.

3.2 Sleep Cycle and UTC Offset Determination

The following procedure is used to identify the UTC offset in the geographic area from which the creation times originate. Given a set of creation times:

⁴ download.geonames.org/export/dump/timeZones.txt.

1. Creation times to Time Distribution:

- (a) The hour from each creation time is used to generate a histogram, with 24 bins corresponding to 24 h.
- (b) Time distribution refers to a normalized histogram; $f(t)$ used to denote the relative frequency of creation times within t th hour.

2. Preprocessing:

- (a) The 24-h time distribution is duplicated to generate a 48-h distribution.
- (b) The distribution is smoothed by computing the moving average of $n = 5$ consecutive points.

3. Sleep cycle identification:

- (a) If there are four intersection points (between $f(t)$ and the $p = 33\%$), per 48 h, the sleep cycle is identified as a single continuous segment between two consecutive intersection points where the first has a negative and the second a positive slope.
- (b) A quadratic function is fitted over sleep cycle: $f(t) = c_0 + c_1 \times t + c_2 \times t^2$. If $c_2 > 0$, its minimum is considered to be the group's *Potential Sleep Time (PST)*, subtracting 24 if needed, so that $PST \in [0, 24)$.

4. UTC offset computation:

- (a) Given a $PST \geq 14$ the transformation $PST-24$ is applied to transform PST from $[0, 24)$ range to the UTC range $[-10, +14)$.
- (b) Linear regression on known data is used to express the UTC offset as a linear function of PST, using Eq. (2) at the end of this section based on Fig. 3.

As an illustration, Fig. 1 shows the $f(t)$ formed from account creation times corresponding to users associated with locations (a) 'london' and (b) 'losangelesca'. The data (blue lines) is noisy, and to achieve smoothness we compute moving averages (with $n = 5$ consecutive points), depicted by green lines.

It is assumed that the regions around the minima (in the smoothed curve) correspond to a nocturnal period when many residents of the region sleep, and hence are not active on social media. This region, expected to be an 8-h period (a third of the 24-h cycle) is identified using the threshold $p = 33\%$ in Fig. 1). The portion of the smoothed curve below the threshold can be approximated by a quadratic function. Minimum of the quadratic used to predict the UTC offset; confidence in which increases with the coefficient of determination R^2 and the magnitude of the power coefficient c_2 (c_2 close to zero associated with a flat like sleep cycle with not as clear a minimum). We hence record (i) the predicted UTC offset, (ii) the power coefficient c_2 , and (iii) the coefficient of determination R^2 .

The next subsection addresses the selection of parameters for the moving average n and the percentile p threshold, and describes the linear regression leading to the computation of UTC.

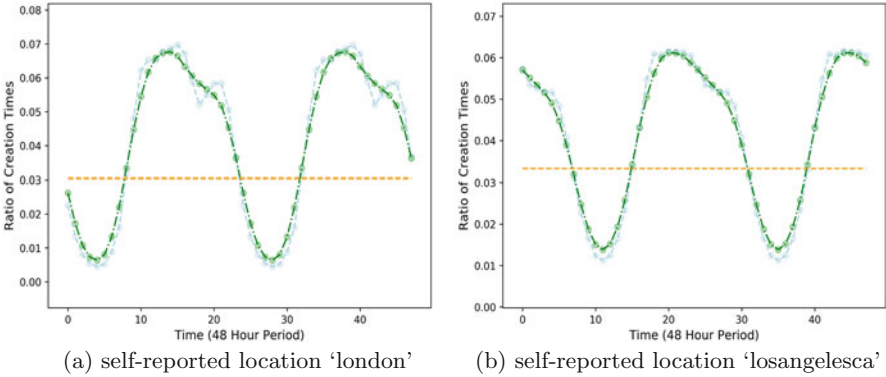


Fig. 1 Normalized 48-h histograms using creation times of users from (a) 'london' and (b) 'losangelesca' are shown. The blue curve shows the original time distribution, and the green curve represents the moving average (with $n = 5$). The orange line corresponds to the threshold below which the potential sleep cycle is identified from the green curve. The mins between the two charts are 7–9 h apart matching expectation in that the time difference between the two locations is 8 h

3.3 Parameter Determination

Instead of using the entire $|G|$ creation times of the group, we use a method akin to bootstrapping [27]). Random samples of size M are drawn from G , N times, and for each sample, the PST is calculated. Over N trials, the average PST is denoted $\mu_G(PST)$, and $\sigma_G(PST)$ denotes the standard deviation.

These estimates depend on the choices of the sample size M , the number of samples N , the size of moving average window n , and the sleep cycle threshold percentile p . We performed multiple experiments, with values of $M = [100, 250, 500, 1000]$, $N = 100$, $n = [1, 2, \dots, 7, 8]$ and $p = [20, 25, 30, 33, 35, 40, 45]$. Linear regression was performed for PST vs. UTC^L using least squares estimation, as shown in Fig. 3. To measure the performance of selected values of the parameters *Recall*, *Precision*, and *F1* measures were calculated; where,

$$Recall = \frac{\# \text{ of user groups where PST-estimate calculated}}{\text{the number of user groups}},$$

$$Precision = \frac{\# \text{ of correct UTC predictions}}{\# \text{ of UTC predictions}},$$

and

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

Predictions that were more than $t_1 = 0.5$ away from UTC^L were marked as incorrect. The following observations emerge from Fig. 2:

- Figure 2a shows F1 for different values of p and n for $M = 250$ and $t_1 = 0.5$ over all groups in the UTC offset dataset. It can be seen that the best performance with F1 = 68.14% is achieved using $p = 33$ and $n = 5$.
- Figure 2b confirms that $p = 33$ is the best performing using precision for four different values of M . This value of p is also an intuitive choice because, as mentioned earlier, about a third of the 24-h period is expected to be devoted to sleep. Smaller percentile ($p < 30$) reduces the associated sleeping cycle and it is harder to fit a parabola and to get a good UTC offset prediction. On the other hand, if p is too high ($p \geq 40$) then points that are outside of the sleeping cycle will be incorporated causing the performance to suffer.
- Figure 2c shows that $n \in [2, 5]$ exhibit high precision for all values of M . From this figure, we conclude that any choice of $n \in [2, 5]$ is reasonable to smooth out

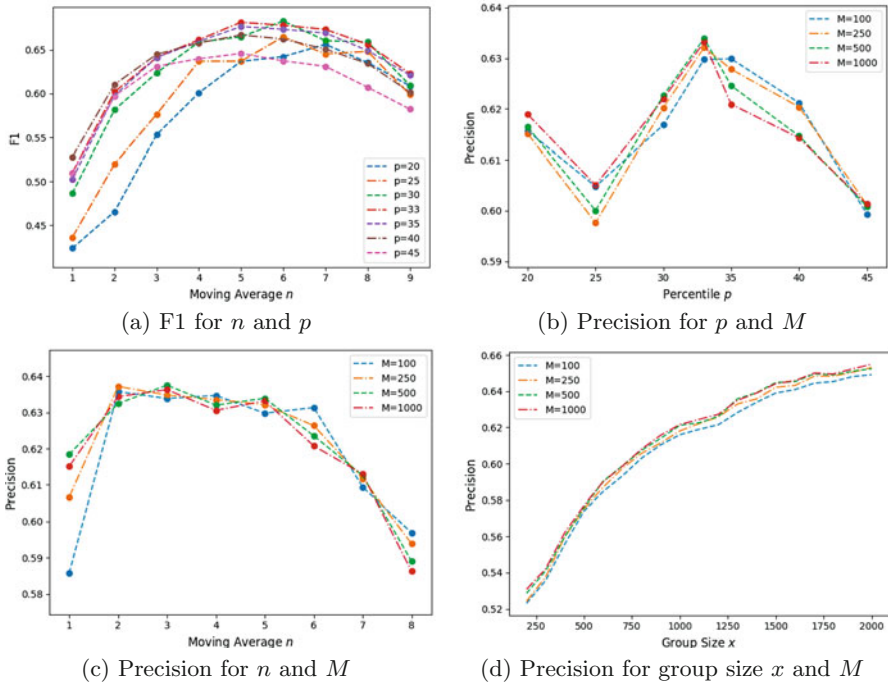


Fig. 2 Variation of ability to predict UTC^L ($t_1 = 0.5$) with parameter values: (a) F1 vs. moving average window width n , for different values of percentile p , fixing $M = 250$; (b) Precision vs. percentile p , for different sample sizes M , fixing $n = 5$ which yielded the best F1 score; (c) Precision vs. n for different values of M , fixing $p = 33$ which yielded the best F1 score; and (d) Precision vs. group size x for different values of M , using sampling with replacement, and fixing $p = 33$ and $n = 5$

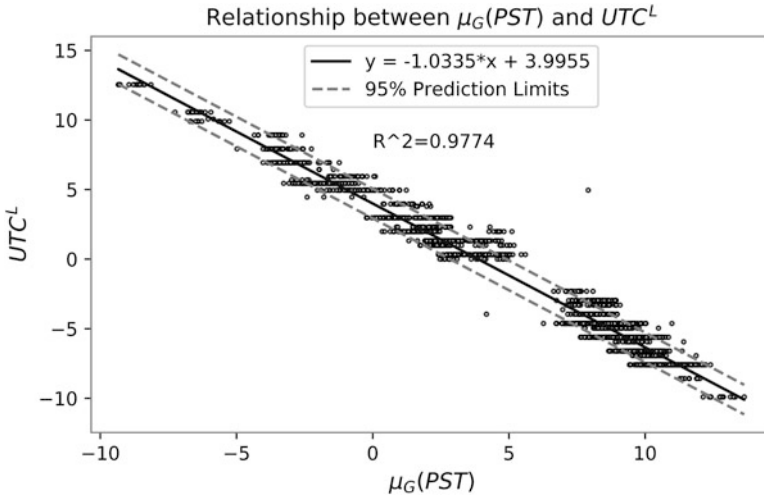


Fig. 3 Result of linear regression performed on data points with known geolocation, plotting UTC^L against $\mu_G(PST)$, with $M = 250$, $n = 5$, $p = 33$, and group size exceeding 1000

irregularities, preserve high precision, but is not too high to delete the sleep cycle from the time distribution. However, considering both, the precision and F1, we conclude that $n = 5$ is the best choice.

- When using sample size M the group size needed to be at least M because we have used sampling without replacement. Sampling with replacement allows to better understand whether improvement comes from a bigger sample size or a bigger group size. Figure 2d shows performance for sampling with replacement across different M values as the group size increases (using $n = 5$ and $p = 33$). We conclude that performance is not affected by M , although performance improves with group size.

The plot in Fig. 3 uses $M = 250$, $n = 5$, $p = 33$, and group size equal to at least 1000. Using these parameters the relationship between predicted PST and actual UTC is shown. A linear relationship can clearly be observed, using $UTC^P = -1.0335 \times PST + 3.9955$ with overall $R^2 = 0.9774$; this is approximated as follows:

$$UTC^P = -1.0 \times PST + 4.0 \quad (2)$$

4 Temporal Analysis of Message Traffic Data

In this section, we illustrate that time-based features can be used for associating persons and topics with a geographic area. The time-based approach is confirmed using message traffic with coordinates.

Our focus is on understanding the spatiotemporal aspects of the Twitter social graph, connecting senders of messages and users mentioned in the messages. We explore the geographical distribution of senders of messages who mention an individual, thereby evaluating the extent to which an influencer (mentioned in the messages) has global influence. This is often accomplished by analyzing message traffic data, since the full follower-followee graph cannot be directly collected due to limitations imposed by the free Twitter API. Messages with coordinates and place mentions or self-reported locations of the users can be used to filter out users that are near a specific geographic area; in this manner, influential individuals and communities belonging to a certain geographic area can be identified.

4.1 Message Traffic Dataset

We collected five days of message traffic data in the first week of December 2020 for a total of 18.67 million messages. This dataset is denoted as D_{mess} . Preprocessing consisted of turning each message to lowercase and tokenizing using NLTK library's TweetTokenizer. For each message, the hour was extracted from its creation time. Each token was associated with a set of hours from the set of messages in which the token appears. Tokens that were at least three characters in length and appeared in over 500 messages were retained, resulting in a total of 23,747 tokens.

Messages that contain location coordinates provide ground truth against which we can evaluate UTC-based predictions. Such messages comprised only 0.71% of all messages in our dataset, consistent with other literature suggesting that the number is less than 1% [16]. In our dataset, there were 6632 messages with point coordinates and 126,765 messages with a place coordinate (bounding box).

Among 23,747 tokens, as many as 20,252 were contained in at least one message with coordinates. For each token, we record the number of messages that came from the Americas (longitude ≤ -25), Europe/Africa ($-25 < \text{longitude} \leq 65$), and Asia/Oceania (longitude > 65). For coordinates specified using a bounding box, both the longitude components had to be associated with the same region. A token was assigned a label based on the region which captured the biggest ratio of messages. Among the 20,252 tokens with coordinate information, we found that 11,955 were associated with the Americas, 4991 with Europe/Africa, and 3306 with Asia/Oceania. Table 2 shows examples of ground truth generated in this fashion that contain topic or person mentions (NA_SA = Americas, AF_EUR = Europe/Africa, and AS_OC = Asia/Oceania).

Table 2 Token labels using messages with geolocation tags

| Token | Label | NA_SA | AF_EUR | AS_OC | Total | Ratio for label |
|------------------|--------|-------|--------|-------|-------|-----------------|
| @realdonaldtrump | NA_SA | 537 | 47 | 19 | 603 | 0.89 |
| @joebiden | NA_SA | 142 | 14 | 6 | 162 | 0.88 |
| #oath4ssr | AS_OC | 18 | 2 | 30 | 50 | 0.6 |
| @narendramodi | AS_OC | 1 | 0 | 47 | 48 | 0.98 |
| #gfvip | AF_EUR | 0 | 35 | 0 | 35 | 1 |
| @pmoindia | AS_OC | 0 | 1 | 33 | 34 | 0.97 |
| @thehill | NA_SA | 25 | 2 | 0 | 27 | 0.93 |
| @jairbolsonaro | NA_SA | 27 | 0 | 0 | 27 | 1 |
| @nytimes | NA_SA | 21 | 3 | 3 | 27 | 0.78 |
| @llinwood | NA_SA | 24 | 1 | 1 | 26 | 0.92 |

4.2 Predicting Region of Token

We extracted the hours (from the creation time) associated with all messages in which each token appears. The set of hours was used to obtain a time distribution and corresponding: (i) predicted UTC offset, (ii) coefficient of the quadratic term, c_2 , and (iii) coefficient of determination R^2 (using the approach in Sect. 3.2). As before, a large value of R^2 implies greater confidence in the fitted polynomial, and a large c_2 indicates greater localization of influence.

The NLTK library contains a list of stop-words, such as ‘the’ and ‘has’, which are used worldwide. Their temporal distributions are flat and associated c_2 is close to zero. For example, we found that for stop-words the largest c_2 was smaller than 0.001. To further refine our dataset, we considered $c_2 \geq 0.001$. The result was that not only stop-words but other global topics and persons such as *#covid19* and *@YouTube* were removed.

Out of 23,747 tokens in the dataset, 16,744 contained a sleep cycle that could be used to predict a UTC offset. Based on predicted UTC offset the token was assigned one of three regions: (i) North and South America ($UTC \leq -2$), (ii) Europe and Africa ($-2 < UTC \leq 4$), and (iii) Asia and Oceania ($UTC > 4$). The number of tokens associated with each region was (i) 9618, (ii) 3012, and (iii) 4114 respectively. Of the UTC predictions, 15,087 had $R^2 \geq 0.85$ of which 8487 had $c_2 \geq 0.001$. Among these 8487 higher confidence predictions 4135, 1416, and 2936 belonged to each region, respectively. As an illustration, Fig. 4 shows the top fifty topic (#) and person (@) mentions in a word cloud.

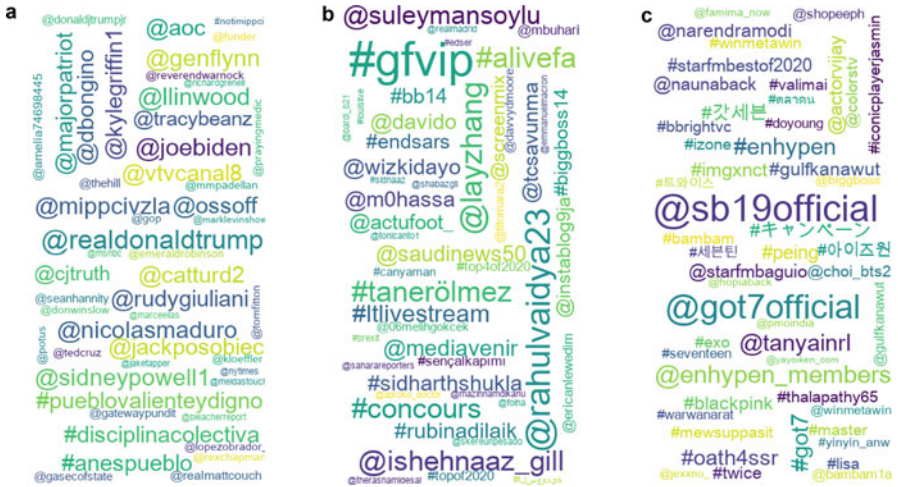


Fig. 4 Top trending tokens (topics and persons) for the (a) North and South America, (b) Europe and Africa, and (c) Asia and Oceania. These were identified using UTC prediction from time curve over message creation times containing the token

4.3 Evaluation

For each token, one of the three regions using UTC prediction is compared with the ground truth, and the accuracy of prediction is recorded as the ratio of correct versus total predictions, for each region.

Table 3 shows the results for the three regions. The first column shows the type of restrictions placed on a token, based on (i) R^2 of the polynomial over corresponding sleep cycle, (ii) power coefficient c_2 from polynomial, (iii) number of minimum messages, x , used to build ground truth, and (iv) whether a collection is limited to persons/topics (@/#). The accuracy of predictions for each region is shown in columns 3–5 (the respective number of predictions per region is shown in the second column). The final column is the overall accuracy of all predictions.

Table 3 Performance over message traffic

| Restriction | Predictions | NA_SA | AF_EUR | AS_OC | Overall |
|--|------------------|-------|--------|-------|---------|
| None | 9271, 2825, 2602 | 94.56 | 76.14 | 87.78 | 89.82 |
| $R^2 \geq 0.85$ | 8611, 2426, 2360 | 95.4 | 80.3 | 89.58 | 91.64 |
| $R^2 \geq 0.85, c_2 \geq 0.001$ | 4008, 1327, 1976 | 98.6 | 89.9 | 95.29 | 96.13 |
| $R^2 \geq 0.85, c_2 \geq 0.001, x \geq 5$ | 3304, 810, 967 | 99.21 | 91.98 | 98.24 | 97.87 |
| $R^2 \geq 0.85, c_2 \geq 0.001, x \geq 10$ | 2270, 380, 533 | 99.69 | 94.47 | 98.69 | 98.9 |
| $R^2 \geq 0.85, c_2 \geq 0.001, @/#$ | 261, 61, 137 | 98.08 | 81.97 | 97.08 | 95.64 |

Table 3 illustrates that the approach using temporal distribution is successful. The first row, with no restriction, illustrates that if a sleep cycle is found and a UTC prediction is made it generally has good accuracy. The accuracy is high, particularly for those tokens which have ground truth assembled from more messages (larger x) and with high confidence UTC predictions (high R^2 and c_2).

About 13% of the tokens that were labeled using UTC did not have any message traffic with coordinates. A bigger collection could be explored, but there is reason to think that some tokens just won't get a geo-tag assigned. Fewer than 1% of messages contained geo-tags, and 38% of the tokens had fewer than 5 geo-tagged data points; a prediction based on such a small sample is not made with high confidence. On the other hand, time is available for all messages and each token appeared in at least 500 messages giving us greater confidence in the corresponding time distribution. This illustrates the usefulness of our approach.

Another alternative would be to utilize the self-reported locations of the users that wrote the messages, but this would require a complex geocoding solution that can handle the different ways persons refer to locations in various languages. Using time distributions is hence a better solution for quickly understanding important keywords in message traffic as they pertain to a geographic region of interest.

4.4 Comparison Against Baseline Based on Google Trends

In a recent paper, Zola et. al. [28] attempt to estimate worldwide Twitter user locations without relying on geolocation target labels (no geotagged tweets or user location profiles and no access to geographic dictionaries). Their dataset consisted of 744,830 tweets written by 3298 users from 54 countries. For each user, their approach was to: (i) collect all of the user's messages, (ii) use Part of Speech (POS) processing to identify nouns, (iii) utilize City Google Trends to associate a noun with a set of weighed city names, (iv) geocode each city name to get its latitude and longitude coordinates, and (v) apply clustering to identify the most probable centroid from coordinates and weights associated with each city. Because no geoinformation is used, the problem is more complex; their approach correctly predicts the ground truth city locations of 15%, 23%, 39%, 58%, 70%, 82% of the users for tolerance distances of 250, 500, 1000, 2000, 4000, and 10,000km. Our method also does not utilize any geoinformation, relying only on creation times as the feature, and hence it was appropriate to compare against this baseline based on Google Trends.

From examining the JSON response from Google Trends (also using the Python Pytrends library) we find that Google, at the 'City' resolution, does return coordinates for each city; hence it is possible to combine steps (iii) and (iv) from above. Other differences are that our focus was on all tokens (not just nouns) and we already had three predefined regions that the world is broken up into (so the performance was judged based on how well a region is predicted, as was done in previous subsection).

The baseline uses Google Trends to identify a set of cities for each token: set A from the Americas (longitude ≤ -25), set B from Europe/Africa ($-25 < \text{longitude} \leq 65$), and set C from Asia/Oceania (longitude > 65). Next for each set of cities in A, B, C the cumulative score across the city weights is recorded (where each weight is from 0 to 100). A token is assigned to a region that has the biggest cumulative score.

Because our problem involves large geographic regions it is also appropriate to utilize the ‘Country’ resolution vs. only ‘City’. Google Trends provides results over a predefined time in the past: past 1 h, 4 h, 1 day, 7 days, 90 days, 1 year, 5 years, all (these are predefined and Google Trends does not allow one to enter a custom date range). Timeframe chosen may have a big impact on the rankings, for example for @realdonaldtrump the top country is Kenya using 1-month, Canada using 3-month, and USA using 1-year timeframe (for collection during April of 2020). Our analysis utilized the default 1-year timeframe.

Google Trends has a limit of around 1440 daily requests. In our evaluation, we have focused on 3183 tokens that had $R^2 \geq 0.85$, $c_2 \geq 0.001$, $x \geq 10$ and 459 tokens that are limited to persons/topics (see the last two rows of Table 3). When we tried to focus on 459 tokens starting with @/#, only 9/459 had results at city and 241/459 had results at the country resolution. This illustrates that Google does not have enough information for trend analysis over these popular Twitter concepts (as these are not as commonly used in Google Search).

Table 4 show the results for the three regions at city and country levels using the 3183 tokens. The first column shows the type of restrictions placed on results from Google Trends ranking. Restrictions considered were using the ranked location with: (i) the highest weight, (ii) the top three weights, (iii) weights ≥ 50 , and (iv) using all. The rest of Table 4 is structured the same as Table 3.

Google Trends has information on most of the tokens with 3039/3183 at the city resolution and 3157/3183 at the country resolution. At the city resolution, it is seen that as more cities are considered the precision is gradually going up i.e. performance using just the top city is the worst. On the contrary, the performance using Country Google Trends has better overall performance when using only the

Table 4 Time-based approach vs. baseline based on Google Trends

| Restriction | Predictions | NA_SA | AF_EUR | AS_OC | Overall |
|--------------------------------|----------------|-------|--------|-------|---------|
| City using top 1 | 2188, 360, 491 | 78.29 | 84.72 | 83.1 | 79.83 |
| City using top 3 | 2188, 360, 491 | 80.94 | 87.5 | 84.73 | 82.33 |
| City using weight ≥ 50 | 2188, 360, 491 | 84.32 | 90.83 | 84.11 | 85.06 |
| City all | 2188, 360, 491 | 90.81 | 94.17 | 89 | 90.92 |
| Country using top 1 | 2267, 365, 525 | 64.49 | 76.16 | 88.76 | 69.88 |
| Country using top 3 | 2267, 365, 525 | 55.32 | 75.07 | 88.38 | 63.1 |
| Country using weight ≥ 50 | 2267, 365, 525 | 59.51 | 78.9 | 87.62 | 66.42 |
| Country all | 2267, 365, 525 | 56.64 | 94.79 | 80.76 | 65.06 |
| Our approach using time | 2270, 380, 533 | 99.69 | 94.47 | 98.69 | 98.9 |

top Country. This could be because there are a lot of separate countries in Europe and Africa continent and as a result, this region tends to be heavily favored when focusing on all countries. Our proposed time-based method performs the best with 98.9% overall precision vs. the best results via Country Google Trends at 69.88% and City Google Trends at 90.92%.

5 Evolving Popularity: Inferring Daily Changes in Number of Followers

It is important to understand how an individual’s influence changes with time; this can help predict future influence as well. To predict the future one must first understand the past. In the context of Twitter, the corresponding problem involves estimating the rate at which an influencer has gained their existing followers over a given time period. In this section, we propose a novel algorithm to address this problem, using the account creation times of an influencer’s followers.

To find the number of followers an influencer has gained on a daily basis (i.e., within a span of 24h) during a period of d days, one would need $d + 1$ daily collections. Since this is a time-expensive proposition and because Twitter API doesn’t allow one to go back in time, we propose an alternative method for approximating daily gains for an influencer, and compare it with an approach based on Meeder, et al. [29].

5.1 Dataset: Stable, Global, Growing Influencers

Each Twitter user’s profile contains the number of followers that the user currently has. By collecting user’s profile multiple times we can get a sense for how the number of followers is changing. Let $\psi(i, t)$ represent the number of followers of influencer i at time t . Let $\psi(i, t_0, t_1) = \psi(i, t_1) - \psi(i, t_0)$ represents the number of new followers i gains during the time interval $[t_0, t_1]$; the number of followers stated in influencer’s profile at t_1 minus the number of followers stated in influencer’s profile at t_0 .

Twitter keeps track of popular celebrity users via *@verified*. There are over 300K verified influencers as of this writing. Our focus is on global stable verified influencers that continue to gain followers; to this end, we collected data on influencers that met the following criteria. The influencer:

1. having greater than a million existing followers, i.e., $\psi(i, t) > 10^6$;
2. gaining at least a thousand followers within 24 h, i.e., $\psi(i, t_0, t_1) \geq 10^3$ where t_0 and t_1 are 24 h apart;
3. the gain in the number of followers is less than 1% of the overall existing follower base, i.e., $\psi(i, t_0, t_1) \leq (0.01 \times \psi(i, t_0))$;

We ensured that the above criteria were met over three $\psi(i, t_0, t_1)$ collections performed in December 2020. Let U_0 contain all verified Twitter accounts. For each influencer i in U_0 we computed $\psi(i, t_0 = d_0, t_1 = d_1)$ where d_0 and d_1 are 24-h apart. Influencers that met the three criteria from above form the set U_1 . For each influencer i in set U_1 we ensured that the three criteria were again met using $\psi(i, t_0 = d_2, t_1 = d_3)$ where d_2 and d_3 are 24-h apart to obtain set U_2 . The process is repeated again using $\psi(i, t_0 = d_4, t_1 = d_5)$ yielding the final set U_3 consisting of 600 influencers.

5.1.1 Data Collected for Each Influencer

The data collected is used to illustrate that $\psi(i, t_0, t_1)$'s, where t_0 and t_1 are 24 h apart, can be predicted using the account creation times. Using an instance of the Twitter API we collected the first 50K followers for each influencer $i \in U_3$. Twitter API instance is used to record the profile metadata and store them to $allProfile = \{allprofile(t, i) : i \in U_3, t \text{ refers to the time of collection}\}$. Profile collection is repeated every 5 min with the list of collection times given by PC .

Another Twitter API instance collects followers. The follower collection, unlike profile metadata, cannot be performed quickly across all influencers. The time when influencer i 's followers are collected is recorded as $Followers_t(i)$.

Once the followers for all influencers are collected: given an influencer i , PC is used to find a the closest time to $Followers_t(i)$ (which we refer to as t_1) and to $Followers_t(i) - 24 \text{ h}$ (which we refer to as t_0). Recall $\psi(i, t_0, t_1) = \psi(i, t_1) - \psi(i, t_0)$, in this case $\psi(i, t_0, t_1) = allProfile(t_1, i) - allProfile(t_0, i)$.

In this way, we have $\psi(i, t_0, t_1)$ over the same period that the followers were collected for all users in U_3 . In the rest of the paper, we refer to $\psi(i, t_0, t_1)$ over all users in U_3 as the actual 24 h follower gain, a_{24} .

The followers and the a_{24} over all users in U_3 make up our dataset that is denoted as D_{600} . Table 5 shows ten influencers from our dataset ordered by the highest a_{24} .

Table 5 Follower gain for selected influencers over 24 h

| Influencer | Follower at t_0 | Follower at t_1 | a24 = Gain |
|--------------|-------------------|-------------------|------------|
| joebiden | 22,009,684 | 22,057,780 | 48,096 |
| bts_twt | 31,718,727 | 31,766,383 | 47,656 |
| bts_bighit | 26,238,967 | 26,280,220 | 41,253 |
| arianagrande | 80,458,070 | 80,494,729 | 36,659 |
| elonmusk | 41,178,206 | 41,208,685 | 30,479 |
| bighitent | 18,527,632 | 18,553,608 | 25,976 |
| kamalaharris | 13,553,348 | 13,577,323 | 23,975 |
| narendramodi | 64,532,998 | 64,556,393 | 23,395 |
| iamcardib | 15,913,538 | 15,935,631 | 22,093 |
| nasa | 42,743,031 | 42,763,315 | 20,284 |

5.2 An Algorithm to Estimate Follower Gain

Meeder et al. [29] observed that the followers of an influencer are returned by Twitter in a list that is in the order of following time i.e. most recent follower first.

Dataset D_{600} for each influencer contains 50K followers. For a specific influencer, let $\mathcal{L} = [l_0, l_1, l_2, \dots, l_{49999}]$ be the list of account creation times of its followers. We select the first $24 \times n$ values from this list for generating 24 rows of size n each, denoted as L_1, L_2, \dots, L_{24} . Each L_i is used to generate a time distribution of the account creation times. For example, in Fig. 5, we have plotted 24 such distributions, using $n = 600$ for the influencer @CNN. In this figure, for each distribution, the hour during which the frequency peaks is highlighted in red. We observe that each distribution has a peak and the peak shifts by an hour. Figure 5 is drawn for @CNN but a similar behavior is observed for most global influencers. In the following, we describe the novel algorithm to estimate an influencer's daily follower gains.

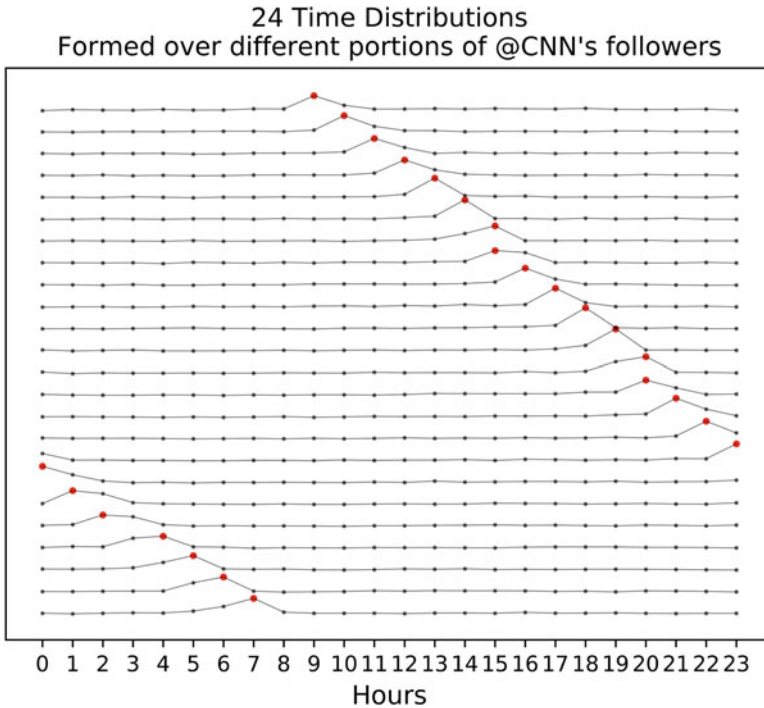


Fig. 5 24 time distributions where each time distribution formed from $n = 600$ followers of @CNN for a total of $24 \times 600 = 14,400$ followers. Distributions are plotted one above the other (L_1, L_2, \dots, L_{24}). For each distribution the hour during which it peaks is highlighted in red

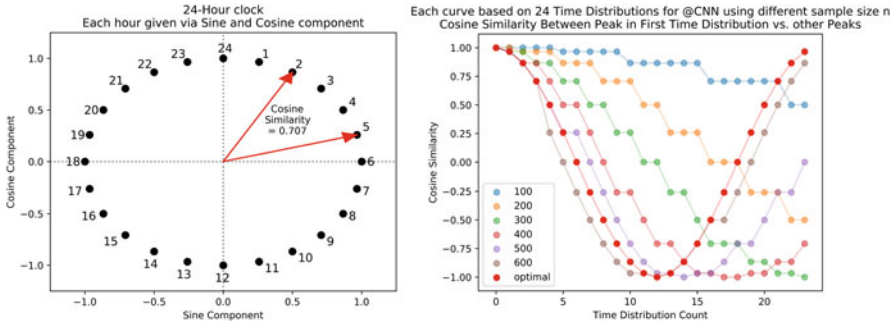


Fig. 6 **Left:** 24-h clock; illustrates the computation of the cosine similarity between two hours. **Right:** Each curve is computed using cosine similarity between the peak time in the first time distribution vs. itself and peak times over remaining 23 time distributions for various n as described in the text (all graphs for @CNN). The red curve is optimal in that the peaks are in the order of the hours on the clock

5.2.1 Representing Cyclical Nature of Time

In the 24-h clock, shown in Fig. 6, each hour can be represented using sine and cosine values of the angle the hour-hand makes with the vertical straight line from the center to the 24th hour. The angle for an hour h in degrees is given as $\frac{360 \times h}{24}$. For example, the angle for 2 O'clock is $\frac{360 \times 2}{24} = 30^\circ$ and 2 O'clock is represented as $(\sin 30^\circ, \cos 30^\circ) = (0.5, 0.866)$. Similarly 5 O'clock is expressed as $(\sin 75^\circ, \cos 75^\circ) = (0.965, 0.258)$. The cosine similarity between $(0.5, 0.866)$ and $(0.965, 0.258)$ is 0.707. If we plot the cosine similarities of (sine, cosine) representation of a specific hour A, with (sine, cosine) representations of hours A, $(A + 1)$, $(A + 2)$, \dots , we obtain a smooth cosine curve (see red plot on the right of Fig. 6). The vector of the above cosine similarities is denoted as V_1 and is called the optimal vector.

Now consider the peak hour in each distribution of @CNN in Fig. 5. If we compute and plot the vector V_2 of cosine similarities between (sine, cosine) representations of these hours with the representation of hour 7 (where the peak occurs in the first distribution), we obtain the brown curve in Fig. 6. Likewise, vectors of cosine similarities resulting from sample sizes given by $n = 100, 200, 300, 400,$ and 500 are shown. The similarity between V_1 and V_2 can be computed using ρ , the Pearson Correlation Coefficient.

We are interested in the size of n that results in temporal distributions that peak exactly one hour apart for all 24 h (or as close to it as possible). For example, in Fig. 6, the curve associated with $n = 600$ is closest to the ideal red curve. The key idea is to try different values of n and calculate the associated V_2 vectors. The vector V_2^* with the highest correlation against V_1 and associated n^* are obtained. The number of followers gained over 24 h is predicted as $p_{24} = 24 \times n^*$. A formal description of the algorithm is provided below.

5.2.2 The Algorithm

Input to the algorithm is the list, \mathcal{L} , of an influencer's followers and the precomputed optimal vector V_1 . Next, n is chosen from a minimum of 10 to a maximum of $\lfloor \frac{|\mathcal{L}|}{24} \rfloor$. For each n , 24 time distributions are generated and from each, the hour during which the time distribution peaks is recorded. V_2 is generated using cosine similarity between the peak hour in first time distribution vs. peaks across all 24 time distributions. V_1 and V_2 are compared using the Pearson Correlation Coefficient ρ . The sample size n^* that resulted in the highest correlation coefficient is returned. The predicted 24 h followers turn over, p_{24} , is given as $24 \times n^*$.

Algorithm 1: *infer24HF(L1):*

```

Input: List L1 of follower account creation times;
Output: Predicted 24 Hour Follower Gain, associated
Pearson Correlation, and number of unique peaks;
bestN, maxP, maxH = 0, 0, 0;
V1 = vector of cosine similarities between
    hour 0 and hours [0, 1, 2, ..., 23];
for n in [10, 15, ..., |L1|/24]:
    Split first 24*n elements of L1 into 24 bins of size n;
    Record the hour with most elements for each of 24 bins;
    V2 = vector of cosine similarities between
        hour in bin 1 and hours in each bin;
    P = Pearson Correlation between V1 and V2;
    if P > maxP:
        bestN = n;
        maxP = P;
        maxH = number of unique peaks across bins;
Return bestN*24, maxP, maxH;
end

```

5.3 Evaluation

For each influencer in D_{600} , we compute p_{24} and compare it to known follower gain a_{24} , using the comparison measure $diff(p_{24}, a_{24}) = \max(p_{24}/a_{24}, a_{24}/p_{24}) - 1$.

Figure 7 shows the scatter plot of p_{24} versus a_{24} for all influencers in D_{600} . The scatter plot is color-coded: green dots represent influencers with $diff \leq 0.25$, and red dots represent large differences with $diff > 1.0$. The Pearson correlation coefficient between p_{24} and a_{24} vectors is $\rho = 0.967$, a high value that shows that the proposed method makes accurate predictions.

We compare our algorithm against two baselines. Meeder et al. [29] provide a method for estimating when a user had followed the influencer. Given a list of followers' account creation times L_1 for influencer i , the follow time for a follower at index j is approximated by $\max(L_1[j :])$ (max gives the most recent account creation time at indices greater than or equal to j). For our problem we are interested in the number of followers gained over 24 h so that the datetime $\max(L_1[j :])$ is as

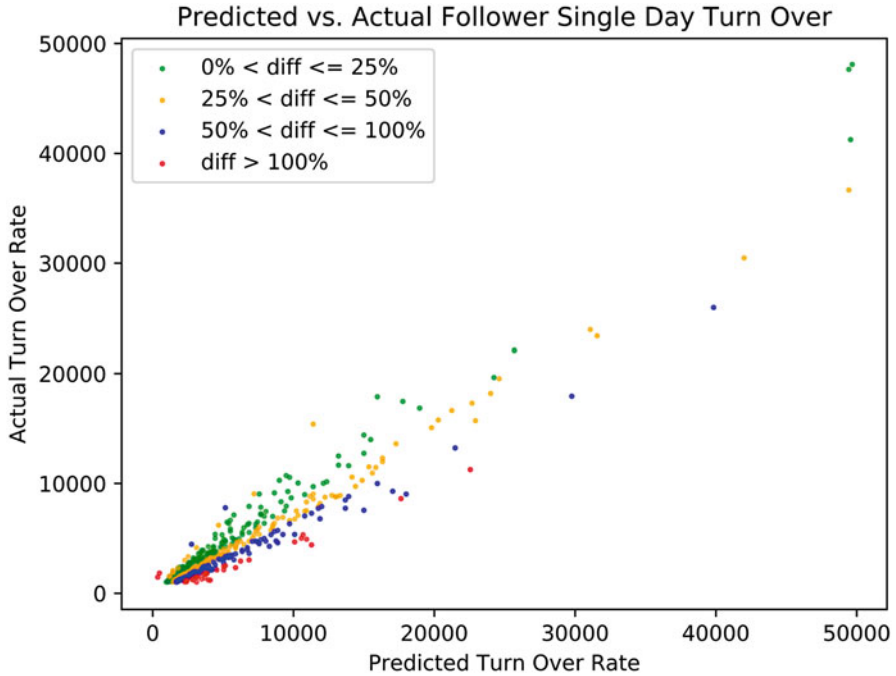


Fig. 7 Scatter plot of inferred vs. actual number of followers gained by 600 influencers over a 24-h time period (Pearson correlation coefficient = 0.967); 238 points (green) differ by <25%, 195 points (orange) differ by <50%, 126 points (blue) differ by <100%, and 41 points (red) differ by $\geq 100\%$

close to the datetime that is 24 h before the follower collection took place (given by $\text{Followers}_t(i)$ minus 24 h). Effectively we are trying to utilize the method proposed by Meeder to estimate the index j that would satisfy this requirement. The method should work for those influencers that are likely to be followed by brand new users immediately after their account creation.

$\text{Followers}_t(i)$ gives time t for influencer i 's followers collection. Let $\mathcal{L}_M[j] = (t - \text{account creation time of the } j^{\text{th}} \text{ follower of the influencer, for } j = 0, 1, \dots)$.

5.3.1 Baseline 1

Traverse the list, \mathcal{L}_M , in reverse order and find the first index j , such that $\mathcal{L}_M[j] \leq 24\text{h}$. If such a j exists, then return $j + 1$, denoted as $p_{24}^{(B1)}$; else return $|\mathcal{L}_M|$.

Table 6 Performance of three algorithms to predict influencers' gains

| Approach | Correlation | Median error | MSE |
|--------------------------------|-------------|--------------|-------|
| Baseline 1 ($p_{24}^{(B1)}$) | 0.962 | 0.620 | 0.665 |
| Baseline 2 ($p_{24}^{(B2)}$) | 0.964 | 0.541 | 0.510 |
| Our | 0.967 | 0.298 | 0.252 |

5.3.2 Baseline 2

For each j , such that $\mathcal{L}_M[j] \geq 24$ h calculate $\frac{\mathcal{L}_M[j]}{j+1}$; Find the minimum ratio, which will approximate the average number of seconds that elapse per new follower; Return $p_{24}^{(B2)} = \frac{86,400}{ratio}$ (86,400 s in 24 h).

As before, we can calculate the correlation coefficient between the vectors of $p_{24}^{(B1)}$ and a_{24} and between $p_{24}^{(B2)}$ and a_{24} over all influencers. In addition, median error and MSE can be computed, where $\text{diff}(\text{predictions}, a_{24})$ is the error that is to be minimized. Table 6 shows how our approach compares against baseline predictions based on these measures. Correlation values of all three approaches are high, with slightly better values obtained by our approach. In terms of median error and MSE, our approach performs much better than the baselines.

5.4 Rationale for Proposed Algorithm and Its Limitations

If we consider a group of users that acted during a specific hour h (such as posting a message or following another), then we are likely to observe a maximum near that same hour in their account creation time distribution. This behavior has been confirmed, as discussed below, by analyzing time distribution for users grouped using the time that they have posted a message.

We utilize the dataset D_{mess} . We take all messages that contain a specific token. For example, for token '@youtube' there were 13704 messages. Next, we separate the messages (containing that token) by the hour of message creation time. In this way, 24 groups of users are formed where each user group is known to have been active during a specific hour (the hour during which the message was generated). For each user group, we construct the account creation time distribution.

Figure 8a shows a heat map for the 24 time distributions generated for token '@youtube'. Notice that a global concept '@youtube' will have a pattern down the diagonal like an Identity Matrix ('@youtube' considered global because $c_2 < 0.001$); the same analysis was performed using stopwords such as 'the' and 'you' and they also observe this pattern. The pattern is due to a unimodal distribution that peaks near the same hour as the hour during which the users were most active in generating the messages. Intuitively if a person had the time for creating their Twitter account in the morning then this person is likely to be active on the Twitter platform during the same morning hours in the future (there is thus a correlation between the account creation times and activity times).

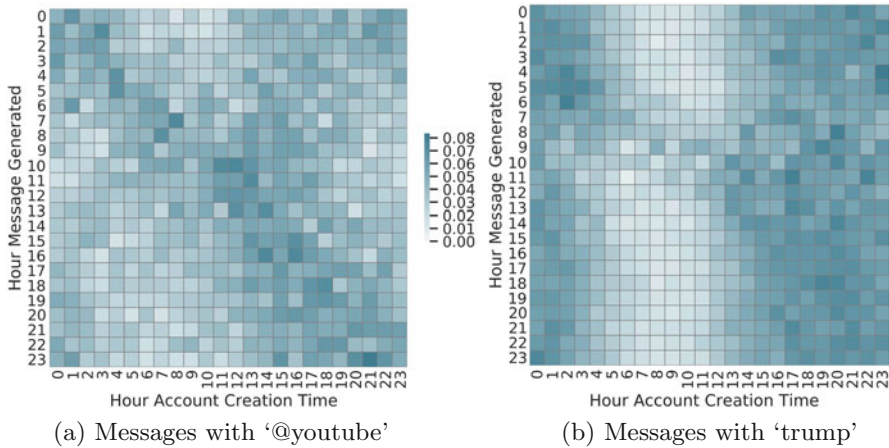


Fig. 8 Heat maps showing 24 time-distributions from users’ account creation times where users are binned by the hour that they generated messages containing tokens: (a) ‘@youtube’ (global) and (b) ‘trump’ (local). For a global token like ‘@youtube’ we see that if a user was active in posting a message during hour h , then the user was likely to have created their account near the same hour h . For token ‘trump’ a sleep cycle is observed (period of inactivity hours 5–11)

The distinction between global and local influencers is illustrated by comparing Fig. 8a vs. b. Figure 8b focuses on a more localized token ‘trump’ that clearly has a period of inactivity, a sleep cycle, during hours 5–11 (token has $c_2 > 0.001$ and during the collection period it was heavily discussed in the Americas).

The concepts observed over message analysis apply to studying the influencer’s followers. We do not know when a user followed an influencer, but because the followers are in sequence of follow time this indicates which followers must have followed earlier on. Algorithm 1 attempts to find a batch of followers of size n that results in a unimodal distribution, which indicates that the followers are likely to have followed the influencer during the same hour as their account creation. When 24 batches, of size n , each peak during a different hour in sequence, it gives confidence that the follower gain around the 24-h time period has been accurately identified (as has been illustrated in Fig. 5 for followers of @CNN and like the Identity Matrix in Fig. 8a).

Algorithm 1, for this reason, is well suited for global influencers that are gaining followers around the clock. In contrast, the heat maps for localized influencers show no strong peaks during some hours of the day. The approach, presented above, also cannot be relied upon for influencers that are gaining no more than 50 followers a day, because the average hourly batch will be too small to generate a meaningful time distribution.

There will be periods during which an influencer gains no followers and even loses followers. We can reason only about followers that the influencer currently has, i.e., we cannot know which followers an influencer might have had in the past. If the influencer has lost many original followers, then the signal in the data will

be obscured by considerable noise; ρ will be small since the peaks will not cover all hours, and the order might not be perfect. Hence we have chosen to focus on influencers that have a large stable following and that are continuing to increase their follower base. It is preferable to pay close attention to ρ and to stop making inferences after ρ goes below some threshold. It is also recommended to compare the modified baseline based on Meeder et al. [29] as an additional check against our method.

5.5 Studying the Evolution of Popularity

To study an influencer's evolution of popularity we need to find how many followers the influencer has gained over multiple days. In an earlier subsection, we have shown that we can estimate an influencer's follower gain over past 24 h. The same technique can be repeatedly applied to study the gains over a longer time span.

To understand the evolution of an influencer's popularity, we first find its followers' creation time list, L_t , obtained at time t . Unlike the list in the previous section that contained only 50K followers, this list consists of all available followers of the influencer.

Say we have an influencer with ten million followers. We could send the whole list to Algorithm 1, but it is not reasonable for the influencer to have gained ten million followers in 1 day, and so to reduce computation we send a smaller more reasonable list. The feature $wSize$ sets the threshold for the maximum number of followers to send to Algorithm 1 (this threshold can be increased or decreased based on influencer's popularity).

Using the first $wSize$ followers between indices $[0, wSize - 1]$ of L_t , Algorithm 1 calculates the number of followers gained between t and $t - 1$, denoted as p_{24}^t . The next $wSize$ followers between indices: $[p_{24}^t, wSize + p_{24}^t - 1]$ will calculate p_{24}^{t-1} (gain between $t - 1$ and $t - 2$). The next $wSize$ followers between indices: $[p_{24}^t + p_{24}^{t-1}, wSize + p_{24}^t + p_{24}^{t-1} - 1]$ will calculate p_{24}^{t-2} (gain between $t - 2$ and $t - 3$). The daily gains returned as list: $[p_{24}^t, p_{24}^{t-1}, p_{24}^{t-2}, \dots]$ successively going backward in time.

Using this approach with $wSize = 50,000$, Table 7 illustrates the number of followers gained in the last 10 days by two examples of qualitatively different kinds of influencers: *@MrBeastYT* and *@NPR*. The table also contains the associated correlation values (suggesting the degree of confidence), and the maximum number of unique hours captured by the peaks for each calculation from Algorithm 1. We observe that *@MrBeastYT* consistently adds more followers than *@NPR*. *@MrBeastYT* also has higher unique hours and higher correlation, suggesting greater confidence in these predictions. This is reasonable since a more popular influencer will have more hourly followers, and consequently, the time distribution will be formed using more data points.

Table 7 Comparison of numbers of followers gained (p_{24}^{t-d}) over each of 10 days by two influencers, along with correlation values maxP and the number of hours maxH spanned by the followers in each 24-h period

| Day d | @MrBeastYT | | | @NPR | | |
|-------|----------------|-------|------|----------------|-------|------|
| | p_{24}^{t-d} | maxP | maxH | p_{24}^{t-d} | maxP | maxH |
| 0 | 12,480 | 0.989 | 20 | 1680 | 0.759 | 18 |
| 1 | 12,120 | 0.993 | 21 | 1560 | 0.944 | 18 |
| 2 | 14,400 | 0.98 | 22 | 1440 | 0.678 | 20 |
| 3 | 9480 | 0.98 | 22 | 1800 | 0.847 | 16 |
| 4 | 10,800 | 0.989 | 23 | 1440 | 0.91 | 20 |
| 5 | 12,960 | 0.934 | 20 | 1320 | 0.944 | 17 |
| 6 | 10,800 | 0.981 | 21 | 1560 | 0.834 | 19 |
| 7 | 11,520 | 0.978 | 22 | 1680 | 0.972 | 20 |
| 8 | 11,520 | 0.979 | 21 | 1440 | 0.948 | 21 |
| 9 | 10,440 | 0.984 | 22 | 1560 | 0.967 | 21 |

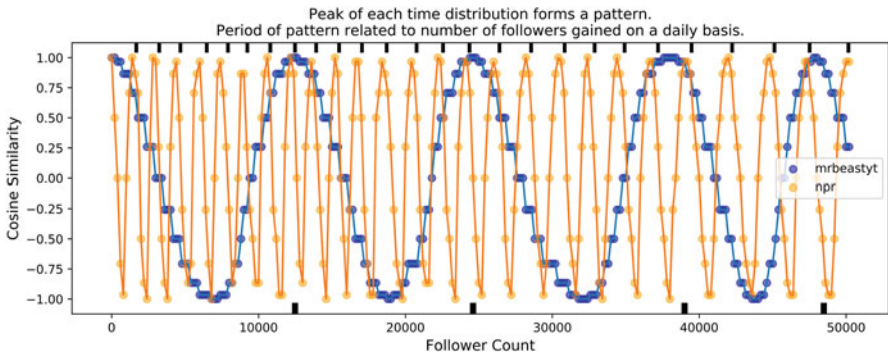


Fig. 9 Daily follower gains from the proposed method are shown as black tick lines on top for @NPR and on the bottom for @MrBeastYT. The cosine similarity curve, as described in text, has a periodicity that predictions from the proposed method can capture. We can thus visually verify that the proposed method is making meaningful predictions going backwards in time beyond a single day

The evolution of popularity for these two influencers can be visualized using Fig. 9, generated by repeatedly taking $n = 200$ followers at a time. The x value corresponds to the index of the last follower in the sample $[n, 2n, 3n, \dots]$. Time distribution is formed over followers using indices $[x - n : x]$ and the hour during which time distribution peaks is recorded. The cosine similarity between the first peak hour vs. the sequence of all peak hours is recorded.

The cosine similarity curve has a periodicity (it starts at 1 goes to -1 and then back to 1). The predicted p_{24}^{t-d} from Table 7 are shown using black tick lines at the top of the chart for @NPR and the bottom for @MrBeastYT. For example for @MrBeastYT the black tick lines appear at $[p_{24}^t = 12480, p_{24}^t + p_{24}^{t-1} = 24600, p_{24}^t + p_{24}^{t-1} + p_{24}^{t-2} = 39000, \dots]$. Visually we can see that the black ticks correspond to the periodicity of the curve for each influencer. In this way, another way to think about our method is in being able to capture the lengths of the periods in Fig. 9, which happen to correspond to the past number of daily followers gained.

6 Global vs. Local Influencer Classifier

In this section, we consider the problem of classifying local versus global influencers. For this, we generate a labeled dataset with 680 local and global influencers. The features are based on sleep cycle analysis (from Sect. 3.2) and peak analysis (from Sect. 5.1.1). The resulting classifier illustrates that the features proposed in this paper are well suited for this task.

6.1 Dataset

The method from [30] is used to generate a list of global and local influencers. Automated Google search queries are utilized to get top Twitter influencers associated with the 100 most populous US cities. The followers of the top influencers are used to generate communities representative of each city. A modified TF-IDF algorithm is used to rank influencers based on whether they have a strong connection to a single city community (local) vs. multiple communities (global). Each influencer was verified manually by reading the influencer’s description and other profile meta-data. In this manner, 680 influencers were identified out of which 558 were local and 122 were global.

6.2 Features

Given a new influencer, we collect the list L_t , of up to 50K followers. Next, Algorithm 1 is applied over L_t to generate features: p_{24} , $\max P$, and $\max H$ (F_0 to F_2 listed below). In the following, the temporal distribution, resulting from the first p_{24} followers in L_t is denoted as $p_{24}Dist$.

1. $F_0 = p_{24}$; if $p_{24} < 500$, $p_{24} = 500$.
2. $F_1 = \max P$: the associated ρ .
3. $F_2 = \max H$; the maximum number of unique hours with peaks.
4. A quadratic is fitted over sleep cycle in $p_{24}Dist$ (as described in Sect. 3.2):

$$F_3 = \begin{cases} c_2 & \text{if sleep cycle exists and quadratic is parabolic} \\ 0 & \text{otherwise.} \end{cases}$$

5. $F_4 = std(p_{24}Dist)$, the standard deviation associated with $p_{24}Dist$.

- 6. F_5 = the fifth Fourier Coefficient (we tried the top 10 Fourier Coefficients⁵ associated with $p24Dist$, but the final classifier did not find others significant. A time distribution with a quadratic will need to be represented using higher order Fourier Coefficients, $F_5 > 0$. Conversely, a simple linear function can be represented using fewer coefficients so that $F_5 = 0$).

6.3 Results: Local Versus Global Classification

We use four families of classifiers:

1. Support Vector Machine (SVM) with the dot, radial, and polynomial kernels,
2. Naïve Bayes,
3. Decision Tree; using information gain with max depth = 5, and
4. Random Forest; the number of trees ≤ 10 , each tree uses information gain with max depth = 5.

Cross-validation with $K = 5$ was employed. Accuracy is averaged over 5 iterations. Decision Tree gave the best results with an average accuracy of $(96.91 \pm 1.08)\%$, followed by the Random Forest $(96.18 \pm 0.86)\%$, and Naïve Bayes $(96.18 \pm 1.27)\%$; SVM performed poorly for all three kernels. The Decision Tree Classifier is shown in Fig. 10.

We used information gain to rank the features. Top four features and their associated weights are: (i) F_1 : 1, (ii) F_4 : 0.983, (iii) F_2 : 0.972, (iv) F_3 : 0.956 (the weight for F_5 : 0.058 so it is not as significant).

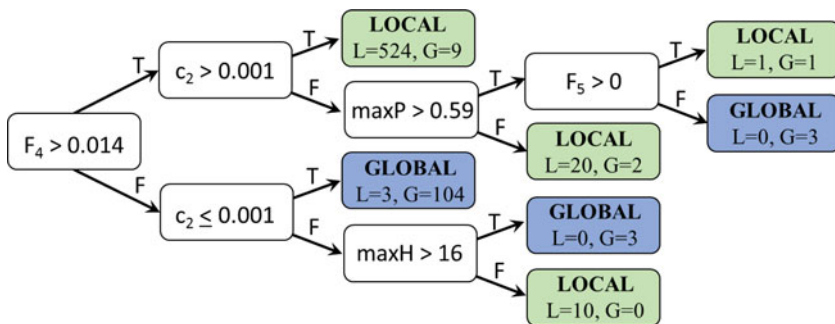


Fig. 10 Decision Tree Classifier for differentiating local vs. global influencers based on the features from account creation times of their followers. The number of local (L) and global (G) influencers predicted using each branch shown for each leaf node

⁵ Complex Fourier transform was used with the SciPy mathematical Python library. The real coefficients corresponding to the cosine terms recorded.

As we have seen in the previous section, a sample of followers from a global influencer can lead to a time distribution that is unimodal, and for this reason, it is important to take a sample determined by Algorithm 1. Algorithm 1 searches for the optimal curve that is achieved if the peaks from time distributions are in sequence and contain all 24 h; if the $\rho(F_1)$ is low and if a small number of hours (F_2) are covered this indicates a local influencer.

If $F_4(\text{std}(p24Dist))$ is low then the spatial distribution is flat and belongs to a global influencer; which is consistent with observations made in the previous sections. The information gain identified that $F_3(c_2)$ less than 0.001 should be the cutoff for a global influencer (this exact value was also confirmed from analysis of stop words using message traffic in Sect. 4). Finally, if the time distribution is represented using only low order Fourier coefficients so that $F_5 = 0$ this means this is more of a flat line simple time distribution associated with a global influencer.

This classifier is intuitive and over the whole dataset achieves $665/680 = 97.79\%$ accuracy. The followers of influencers that the classifier predicts as local can be used for predicting UTC offset related to local expert finding in social networks. While the followers of global influencers can be used for inferring daily follower gains and analyzing how their popularity has evolved.

7 Conclusions

In this paper, we have illustrated an approach for how creation times can be used in time series analysis. The creation times can stem from a group of messages or account creation times. It was illustrated that the distribution of creation times that stem from a single time zone will be approximately parabolic, with a minimum during the night time for that time zone. Regression with a quadratic function can thus be used to predict the UTC offset associated with the time zone. By examining message traffic, this information was utilized to identify trending keywords over multiple geographic areas of interest. In addition, by analyzing the set of followers of any influencer, we showed that this information can be utilized to determine how strongly localized is the range of influence of an influencer. This is useful for Location-Aware Influence Maximization (LAIM) and local expert finding in social networks.

We also illustrated that a follower sample exists such that the peaks from multiple time curves occur in sequence. Analysis of variations of the wave pattern in the distribution of peaks provides information regarding the periodicity with which followers were gained. This is useful for understanding how an influencer's popularity has evolved over time, as well as for inferring link creation times.

Finally, the proposed time-based features were utilized for creating a local vs. global type classifier. The classifier is important because the UTC offset prediction should be applied for local influencers whereas the analysis for how influencer's popularity evolved works for global influencers.

For future work, we plan to study additional techniques from signal processing for processing temporal data and modeling how influencer's popularity evolves. We also would like to study the ranking of local vs. global influencers vs. binary classification as was done in the paper.

Acknowledgments The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

References

1. Yang K-C et al (2020) Scalable and generalizable social bot detection through data selection. In: Proceedings of the AAAI conference on artificial intelligence, vol 34. No. 01
2. Craswell N, de Vries AP, Soboroff I (2005) Overview of the TREC 2005 enterprise track. TREC 5
3. Husain O et al (2019) Expert finding systems: a systematic review. Appl. Sci. 9(20):4250
4. Lappas T, Liu K, Terzi E (2011) A survey of algorithms and systems for expert location in social networks. In: Social network data analytics. Springer, Berlin
5. Page L et al (1999) The PageRank citation ranking: bringing order to the web. Stanford InfoLab, Stanford
6. Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. J ACM 46(5):604–632
7. Weng J et al (2010) Twiterrank: finding topic-sensitive influential twitterers. In: Proceedings of the third ACM international conference on web search and data mining
8. Romero DM et al (2011) Influence and passivity in social media. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin
9. Pal A, Counts S (2011) Identifying topical authorities in microblogs. In: Proceedings of the fourth ACM international conference on web search and data mining
10. Ghosh S et al (2012) Cognos: crowdsourcing search for topic experts in microblogs. In: Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval
11. Cheng Z et al (2014) Who is the barbecue king of Texas? A geo-spatial approach to finding local experts on twitter. In: Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval
12. Li W, Eickhoff C, de Vries, AP (2014) Geo-spatial domain expertise in microblogs. In: European conference on information retrieval. Springer, Cham
13. Li W, Eickhoff C, de Vries AP (2016) Probabilistic local expert retrieval. In: European conference on information retrieval. Springer, Cham
14. Niu W, Liu Z, Caverlee J (2016) On local expert discovery via geo-located crowds, queries, and candidates. ACM Trans Spat Algorithms Syst 2(4):1–24
15. Inkpen D et al (2017) Location detection and disambiguation from Twitter messages. J Intell Inf Syst 49(2):237–253
16. Jurgens D et al (2015) Geolocation prediction in Twitter using social networks: a critical analysis and review of current practice. ICSWM 15:188–197
17. Zheng X, Han J, Sun A (2018) A survey of location prediction on Twitter. IEEE Trans Knowl Data Eng 30(9):1652–1671
18. Graham M, Hale SA, Gaffney D (2014) Where in the world are you? Geolocation and language identification in Twitter. Prof Geogr 66(4):568–578

19. Compton R, Jurgens D, Allen D (2014) Geotagging one hundred million Twitter accounts with total variation minimization. In: 2014 IEEE international conference on big data (Big Data). IEEE, Piscataway
20. Wei H, Sankaranarayanan J, Samet H (2017) Measuring spatial influence of Twitter users by interactions. In: Proceedings of the 1st ACM SIGSPATIAL workshop on analytics for local events and news. ACM, New York
21. Mourad A et al (2019) A practical guide for the effective evaluation of Twitter user geolocation. *ACM Trans Soc Comput* 2(3):1–23
22. Lau JH et al (2017) End-to-end network for twitter geolocation prediction and hashing. Preprint. arXiv:1710.04802
23. Ebrahimi M et al (2018) A unified neural network model for geolocating Twitter users. In: Proceedings of the 22nd conference on computational natural language learning
24. Zannettou S et al (2019) Disinformation warfare: understanding state-sponsored trolls on Twitter and their influence on the web. In: Companion proceedings of the 2019 world wide web conference
25. Kwak H, Chun H, Moon S (2011) Fragile online relationship: a first look at unfollow dynamics in Twitter. In: Proceedings of the SIGCHI conference on human factors in computing systems
26. Kariyaa A et al (2018) Defining and predicting the localness of volunteered geographic information using ground truth data. In: Proceedings of the 2018 CHI conference on human factors in computing systems
27. Efron B, Tibshirani RJ (1998) An introduction to the bootstrap. Chapman & Hall; CRC, London
28. Zola P, Ragno C, Cortez P (2020) A Google Trends spatial clustering approach for a worldwide Twitter user geolocation. *Inf Proces Manag* 57(6):102312
29. Meeder B et al (2011) We know who you followed last summer: inferring social link creation times in Twitter. In: Proceedings of the 20th international conference on world wide web
30. Panasyuk A, Mehrotra KG, Yu ES-L (2019) Automated location-aware influencer evaluation. In: Proceedings of the 3rd international conference on vision, image and signal processing
31. Panasyuk A, Mehrotra KG, Yu ES-L (2020) Improving geocoding of a Twitter user group using their account creation times and languages. In: 2020 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE, Piscataway

Covid-19 and Vaccine Tweet Analysis



Eren Alp, Bedirhan Gergin, Yiğit Ahmet Eraslan, Mert Can Çakmak,
and Reda Alhajj

Abstract Social networks are the most effective instruments for gathering information about people's opinions and perceptions on a variety of subjects and concerns. People spend hours a day on social media to express their ideas, viewpoints, and answers with others. In this chapter, Covid-19 and Vaccine tweets that are taken from two different time manners were analyzed. Python was used to perform experiments on a variety of tweets. After collecting and preprocessing the data, various visualization techniques were used to show the results for most occurred words and sentiment analysis for positivity and negativity of tweets.

Keywords COVID-19 · Vaccine · Tweets · Opinion · Sentiment analysis

1 Introduction

Data that people poured into the internet like reactions and comments on the topics have the potential to reveal valuable insights on human emotions. Thus, the analysis of people's ideas and comments can play a crucial role to understand people's behavior and response in various ways. With the increasing number of microblogs and social media, people have begun to express their opinions on a wide variety of topics on Twitter and other similar platforms. As they are growing and spreading rapidly these tools became more useful to understand and model various events.

In this chapter, a dataset formed of collected tweets from Twitter was used. Twitter contains a large number of short messages created by the users of this microblogging platform. The contents of the messages vary from personal thoughts to public statements.

E. Alp · B. Gergin · Y. A. Eraslan · M. C. Çakmak · R. Alhajj (✉)
Department of Computer Science, University of Calgary, Calgary, AB, Canada

Department of Computer Engineering, Istanbul Medipol University, Istanbul, Turkey

Department of Health Informatics, University of Southern Denmark, Odense, Denmark
e-mail: yaeraslan@st.medipol.edu.tr; bgergin@albany.edu; ralhajj@ucalgary.ca

As a microblogging and social networking website, Twitter has become very popular and has grown rapidly. An increasing number of people are willing to post their opinions on Twitter, which is now considered a valuable online source for opinions. As a result, Twitter sentiment analysis provides a quick and efficient tool to evaluate public opinion for business marketing or social research. In this project sentiment analysis is done about Covid-19 and Vaccine tweets. First word occurrences and some visualizations were used and sentiment analysis was done.

Sentiment is an attitude, thought, or judgement prompted by feeling. Sentiment analysis is the process of determining and measuring the tone, attitude, opinion, and emotional state of responses. More precisely, it is the concept of deciding whether a specific conversation is positive, negative, or neutral. In our study just negativity and positivity of tweets were categorized.

The rest of this chapter is organized as follows. Section 2 covers the related work. Section 3 describes the methodology. Section 4 presents the results. Section 5 is the conclusions.

2 Literature Review

There are works about sentimental analysis, measuring the of the user, and topic modeling. In the Sentiment Analysis and Influence Tracking using Twitter paper [1], the authors mention that how Twitter data is used as a corpus for analysis by the application of sentiment analysis and a study of different algorithms and methods that help to track the influence and impact of a particular user/brand active on the social network. They used Twitter API, Twitter Streaming API, and Twitter Search API for data collection. For analysis preprocessing, techniques such as tokenization, normalization, and part of speech (POS) tagging are used. To determine the influence of the user PeopleRank and TwitterRank algorithms are used. Using these data collection APIs data can be collected from Twitter easily and ranking algorithms can help to calculate the influence of the user.

In the Detecting Real-World Influence Through Twitter paper [2] the authors investigated the issue of detecting the real-life influence of people based on their Twitter account. For the dataset CLEF RepLab, 2014 dataset is used. Social Network Analysis (SNA), Principal Component Analysis (PCA), bag of words, POS, linear classifiers which are Support Vector Machine (SVM) and libLinear, logistic regression, logic boost, multinomial Naïve Bayes are used for determining real-world influence. Since bots are not real influence in the real world this is helpful to detect someone's real influence value. In the Topic Modeling of Twitter Conversations paper [3], the authors presented a way to analyze large amounts of textual data from Twitter conversations efficiently and effectively. Specifically, it was explained how to capture the narratives that people share on Twitter about social events, reduce their complexity, and provide plausible explanations. For this Latent Dirichlet Allocation (LDA) method is used. By using this method, the topics from contexts can be extracted efficiently and effectively.

In the Extracting health-related causality from Twitter messages using natural language processing paper [4], the authors evaluated an approach to extracting causalities from tweets using natural language processing (NLP) techniques. Twitter Streaming API is used for dataset collection. To extract causality, lexicon syntactic relations and NLP pipeline operations which are lemmatizing, POS and dependency parsing are used. Since a good causality relationship sentence results in the good influence of a person when a reader reads that sentence so that this can be used for determining the influence of the user. However, because there are so many distinct methods to express cause and effect relationships in a phrase, it's difficult to keep track of them all.

In the Investigating the Relationship between Trust and Sentiment Agreement in Arab Twitter Users paper [5] the authors proposed a research methodology framework for investigating the relationship between trust and sentiment agreement on Twitter and explain the framework by applying it to a use case from Saudi Arabia. For this, the adaptation of the EigenTrust Algorithm which is the MarkovTrust algorithm is used. Also, surface analysis, deep analysis, and shallow analysis algorithms are used to determine the relationship between trust and sentiment agreement. Since the context and sentiment have been taken into consideration, determining the trust of the user will be more accurate.

In the Influence Analysis of Emotional Behavior and User Relationships Based on Twitter Data paper [6], the authors analyzed the influence of emotional behavior on user relationships based on Twitter data using two dictionaries of emotional words. For the collection of data random sampling, for calculation emotion score Keyword Matching, and the testing Brunner-Munzel test is used. By looking at emotional behaviors the influence of the user can be determined.

To sum up, the related work is summarized in Table 1.

3 Methodology

3.1 Data Collection

Implementing the sentiment algorithm and using it for further steps in the project, as well as a data collection technique. Collecting the data from a social media website was done through a scraper. A scraper is a type of software used to copy content from a website. In this project Snsrape was used for this purpose. Snsrape is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g., the relevant posts.

Shown in Fig. 1 is an example data collection that were taken from Twitter and transformed into csv file.

Table 1 Related works

| Author | Period | Title | Method/remarks |
|---|--------|---|---|
| Pramila M. Chawan | 2012 | Sentiment analysis and influence tracking using Twitter | “Paper makes a sentiment analysis on an entity level; mine people’s idea on specific entities instead of whole tweets, scrutinize each tweet. Use three main features for scoring: followers, mentions and retweets, and lists, these are used for ratio to users” [1] |
| Peiyao Li, Weiliang Zhao, Jian Yang, and Jia Wu | 2019 | CoTrRank: trust evaluation of users and tweets | “Develop a trust ranking approach named CoTrRank. It mainly uses a coupled dual network. By evaluating the coupling affect in users and tweets. Values are derived with their original meanings in different trust spaces. The results of experiment show that the CoTrRank provides better evaluations of the trustworthiness of users and tweets when it is compared with other methods.” [7] |
| Jean-Valère Cossu, Nicolas Dugué, Vincent Labatut | 2015 | Detecting real-world influence through Twitter | “Project analyze Twitter-based features with comparing and allowing to measure the offline effects and influence of users. Look for specific characteristics in twitter that can explain people known to be influential in their real-life.” [2] |
| Son Doan, Elly W. Yang, Sameer S. Tilak, Peter W. Li, Daniel S. Zisook and Manabu Torii | 2018 | Extracting health-related causality from twitter messages using natural language processing | “Causality extraction is done by outputs that are dependency parser of Lexico-syntactic patterns. These techniques were used to help and improve the preciseness of information extraction. Paper shows that dependency parser with lexicon-syntactic relations yields high precision, which is an important feature for big data set mining.” [4] |
| Kiichi Tago and Qun Jin | 2018 | Influence analysis of emotional behaviors and user relationships based on Twitter data | “Paper conduct three different experiments: calculate the average emotion score of a user, calculate the average emotion score using emotional tweets, and calculate the average emotion score using emotional tweets, with not including users of few emotional tweets. Then analyze by Brunner–Munzel test for the influence of emotional behaviors to user relationships. From the result it is understand that a positive user is more active than a negative user for building a user relationship in a specific situation.” [6] |

(continued)

Table 1 (continued)

| Author | Period | Title | Method/remarks |
|---|--------|---|---|
| Areeb Alowisheq and Sarah O Al-Humoud | 2017 | Investigating the relationship between trust and sentiment agreement in Arab Twitter users | “It conducts a research method for identifying the relationship between trust and sentiment for Arab Twitter users.” [5] |
| Younggwe Bae and Hongchul Lee | 2012 | Sentiment analysis of Twitter audiences: measuring the positive or negative influence of popular twitterers | “Paper identify between the positive and negative audiences of popular tweet users. Then, find that the audience are influenced by the sentiments used in the tweets by popular users. Thirdly, from these two findings it develops a positive-negative measurement for influence. Finally, by a Granger causality analysis, it is understood that sentiment change of the audience was related to the real- world sentiment landscape of popular users.” [8] |
| Cano Basave, A. E.; Mazumdar, S. and Ciravegna, F. | 2011 | Social influence analysis in microblogging platforms a topic sensitive based approach | “Paper suggests the use of lexical profiles forming dominant users depending upon the retweet Twitter graph. Establishes a different version of the PageRank algorithm for examining user’s relevance of a retweet connection.” [9] |
| Juyup Sung, Seunghyeon Moon, and Jae-Gil Lee | 2013 | The influence in Twitter: are they really influenced? | “Paper tenders a development of PageRank algorithm, which is InterRank. It regards both relationship and topical similarity among users. It suggests that topical similarity act upon dominance.” [10] |
| Eliana Sanandres, Camilo Madariaga, Raimundo Abello | 2018 | Topic modeling of Twitter conversations | “Paper suggests a technique for topic modeling on Twitter chatting which is Latent Dirichlet Allocation to decide the topics that are talked.” [3] |

(continued)

Table 1 (continued)

| Author | Period | Title | Method/remarks |
|--|---|---|---|
| Liangjie Hong and Brian D. Davison | 2010 Empirical study of topic modeling in Twitter “Paper suggests a solution for normal topic model algorithms that have been used on social media. It proposes that training a topic model with clustered text, it can be achieved better accuracy and preferable performance.” [11] | | |
| Christan Grant, Clint P. George, Chris Jenneisch, and Joseph N. Wilson | 2011 | Online topic modeling for real-time Twitter search | “Paper aims to get the attractive and topical social media entries from the dataset. It uses topic modeling algorithm for examination in the dataset.” [12] |
| Ellas Jonsson, Jake Stolee | 2016 | An evaluation of topic modelling techniques for Twitter | “Paper assesses of different topic modelling algorithms and analyze them by looking their performance on Twitter texts.” [13] |
| Yefeng Ruana, Arjan Durreesia, Lina Alfantoukha | 2018 | Using Twitter Trust Network for Stock Market Analysis | “Paper suggests that using the trust between users on microblogs, this can improve the mutual affinity with financial data in the stock market.” [14] |

3.2 Preprocessing

The preprocessing steps are:

1. Lower Tweets: Text are converted to lowercase.
2. Remove the URLs: Links starting with “http” or “https” or “www” are replaced by empty string.
3. Remove mentions, retweet and hashtags: Words starting with “@”, “#”, “RT” are removed.

| | A | B | tweet |
|----|---------------------|------------------|--|
| 1 | date | user | tweet |
| 2 | 2020-12-25 23:59:58 | Wfdee1 | @LLinWood @TXPSALM55 You got that right I think this whole covid-19 deal was a plan from China and the dems |
| 3 | 2020-12-25 23:59:53 | KennaStevens1 | @TheRickyDavila @doxiedachsie When they were informed of COVID19 early this year they invested in a compar |
| 4 | 2020-12-25 23:59:52 | JemMangler | Short but important thread. #COVID19 #COVID #edchat #iaedchat https://t.co/85Z2nVUMQB |
| 5 | 2020-12-25 23:59:51 | COEmergency | COVID-19 vaccine administered: 63,170 doses #COVID19Colorado https://t.co/lArtrHVaf4 https://t.co/fmWZptXtYA |
| 6 | 2020-12-25 23:59:51 | AntiTotalitabot | @CTVNews inaccurate counting: How COVID-19 Deaths Are Counted https://t.co/gjMyeBNYV2 |
| 7 | 2020-12-25 23:59:46 | TestUser05632971 | Some Passengers Infected After Man Died of COVID-19 on Plane https://t.co/GWXB2OhSHo |
| 8 | 2020-12-25 23:59:45 | Dutch0L | Guerrilla Mask Force Protest Denmark and Germany Covid-19. https://t.co/77dV5b2jfl via @YouTube |
| 9 | 2020-12-25 23:59:43 | rewnowija | News of COVID-19 Vaccine Special Gift of Christmasc™NLCA President https://t.co/6kbA0ae4j |
| 10 | 2020-12-25 23:59:41 | JaimeAnaya | Suspicious grow that nanoparticles in Pfizer's COVID-19 vaccine trigger rare allergic reactions https://t.co/OB |
| 11 | 2020-12-25 23:59:41 | adejoke_mukaila | The spread circumstances substance of covid-19, goes extremely breakout viral in the world. But we have the cau |
| 12 | 2020-12-25 23:59:41 | EarickNG | Covid-19 UK Mutant Strain: Higher Hospitalizations, Deaths Likely, Study Finds - Bloomberg https://t.co/3fFmnrk |
| 13 | 2020-12-25 23:59:38 | julesofmaine | @portlandimber @nirav_mainecc @IMPublicHealth The real truth about Covid-19. Respiratory deaths are no |
| 14 | 2020-12-25 23:59:35 | govfessiss | @JimmyMac2021 @Toronto1880 @fordnation And Peel didnt? |
| 15 | 2020-12-25 23:59:33 | jmonrad | And several @NIH researchers published this piece on "COVID-19 vaccine trial ethics once we have efficacious va |
| 16 | 2020-12-25 23:59:33 | MartyKoekemoer | @AFranzen @Thomas_Binder @tngadd WeK™re becoming preoccupied with Covid19. And Death. And not seei |
| 17 | 2020-12-25 23:59:27 | LiterateLiberal | Yuma Prison Warden Dies From COVID-19 After Dismissing Safety Concerns @crooksandliars https://t.co/bAbit |
| 18 | 2020-12-25 23:59:25 | myraluv2015 | Yes until covid19 is under control. https://t.co/7qaldj08R |
| 19 | 2020-12-25 23:59:23 | PrincetonBoy915 | @Sharlie528 Covid-19? I donK™I know her 8",_fYK |
| 20 | 2020-12-25 23:59:22 | MartyKoekemoer | @LusyNote @Thomas_Binder WeK™re becoming preoccupied with Covid19. And Death. And not seeing the dan |
| 21 | 2020-12-25 23:59:21 | jmonrad | Do these concerns apply here? |
| 22 | 2020-12-25 23:59:18 | NewsThalvisa | Two seafood vendors in Onnut fresh market test positive for Covid - https://t.co/db8aXo6Hu |

Fig. 1 Example Covid-19 tweet data from Snsrape

4. Remove symbols: Emoticons, symbols and pictographs, transport and map symbols, flags, other language characters and dingbats are removed.
5. Remove non alphabet characters: Replacing characters except Digits and Alphabets with a space.
6. Remove consecutive letters three or more: 3 or more consecutive letters are replaced by 2 letters. (eg: "Cooool" to "Cool")
7. Remove punctuations: Punctuations are removed from the sentence since it is not affecting the meaning of the sentence.
8. Remove stopwords: The stopwords are not add much meaning to a sentence.

Shown in Tables 2 and 3 are examples of data and results before and after preprocessing.

Table 2 Tweet examples

| Covid test tweets |
|--|
| @TheRickyDavila @doxiedachsie When they were informed of COVID19 early this year they invested in a company that makes body bags. Why would they care about the nations virus death toll? These two are despicable and must be voted out! Let's go, Georgia! |
| Short but important thread. #COVID19 #COVID #edchat #iaedchat https://t.co/85Z2nVUMQB |
| COVID-19 vaccine administered: 63,170 doses #COVID19Colorado https://t.co/lArtrHVaf4 https://t.co/fmWZptXtYA |
| @CTVNews inaccurate counting: How COVID-19 Deaths Are Counted https://t.co/gjMyeBNYV2 |
| Some Passengers Infected After Man Died of COVID-19 on Plane https://t.co/GWXB2OhSHo |

Table 3 Preprocessed tweet examples

| Preprocessed tweets |
|--|
| Informed covid19 early year invested company makes body bags would care nations virus death toll two despicable must vote let go Georgia |
| Short important thread covid19 covid edchat iaedchat |
| Covid 19 vaccine administered 63 170 doses covid19colorado |
| Inaccurate counting covid 19 deaths counted |
| Passengers infected man died covid 19 plane |

3.3 *Vectorization*

In this part every single word occurrence was counted to fill the word occurrence matrix with words and their number of occurrences. This can be counted as n-grams. An n-gram is a contiguous sequence of n items from a given sample of text. In our case n is equal to 1, which means single word was counted not group of words. After vectorization, we obtained one word occurrence matrix for each csv file.

3.4 *Sentiment Analysis*

There are different types of sentiment analysis types, some of them are; polarity and subjectivity analysis, positivity and negativity analysis, emotion detection. Our project includes positivity and negativity analysis meaning that the result for every tweet is positive or negative. While implementing this, the Naive Bayes Classifier method from TextBlob library in Python was used. The Naive Bayes Classifier is wrapping the same named method from NLTK library in Python and this method classifies movies using a pre-trained model, or the coder can manually train the model with related data. We choose the second approach and trained the model with our labeled tweets dataset, then tested and accuracy was found. Finally, the unlabeled data was given to model and obtained their positivity and negativity values.

3.5 *Visaulization*

The results were all numbers, but they are more meaningful when visualization is good. So, the Matplotlib library of Python was used to draw bar charts, plots, and pie charts. Wordcloud method from TextBlob library was also used for more colorful results for word occurrences.

4 Result and Discussion

In this study, four different Dataset were analyzed. Two datasets from December 2020 about Vaccine (380,000 tweet) and Covid-19(318,000 tweet) and two dataset from January 2021 about Vaccine (500,000 tweet) and Covid-19(212,000 tweet). Accuracy of the sentiment analysis algorithm after training is determined as “0.6”.

In this section, the results of the visualization process and criticism of the results are included. The bar charts and word clouds are the result of vectorization. The table shows us the sentiment analysis result for each dataset.

By considering the datasets collected in December, 2020, occurrences of the most common words related to “Vaccine” in the analyzed tweets are shown in Fig. 2. Occurrences of the most common words about COVID are displayed in Fig. 3. The same two results for the data collected in January 2021 are shown in Figs. 4 and 5, respectively. Comparing Figs. 2 and 3 with Figs. 4 and 5, respectively, it is obvious

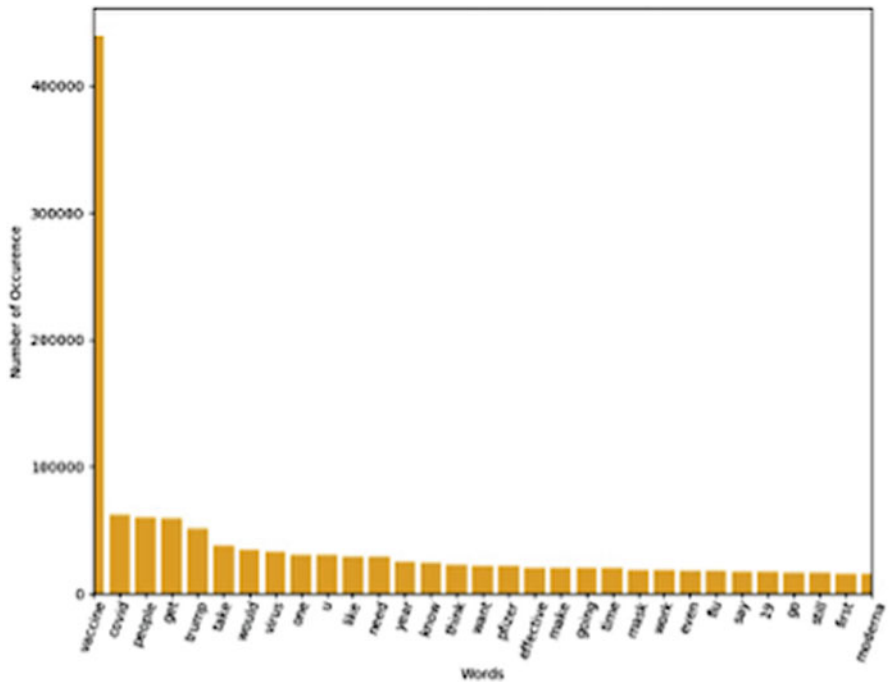


Fig. 2 Most occurred words in tweets about vaccine in December, 2020

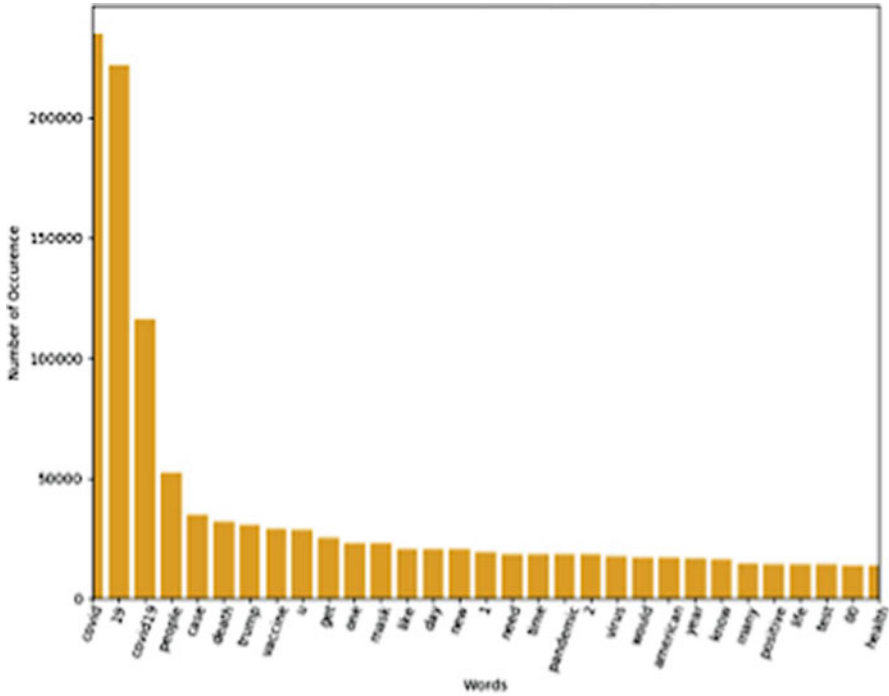


Fig. 3 Most occurred words in tweets about Covid in December, 2020

that the number of occurrences for the common words decreased from December 2020 to January 2021. This may be attributed to various factors, including the following. December is mostly characterized as a vital month with holidays season where people organize a lot of indoor and outdoor activities, travels, etc. On the other hand, January is considered a calm month where people recover from the activities and travel they completed in December. Thus, the drop in the interest in the covid and vaccine can be seen as normal. Further, in January, people are more uninterested in discussing the pandemic after one year of suffering from its health, societal and economic consequences. People tend to be more interested in returning back to normal life style. The most important words discussed during these two periods for “Vaccine” and “Covid” related tweets are reflected in the

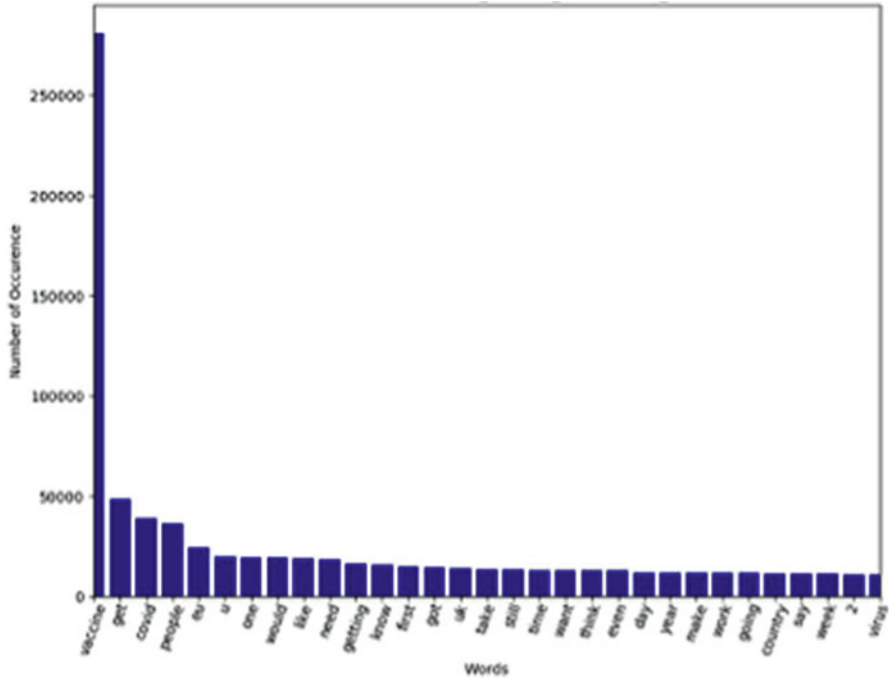


Fig. 4 Most occurred words in tweets about vaccine in January, 2021

word clouds shown in Figs. 6, 7, 8 and 9. The related to sentiments for these two periods (December 2020 and January 2021) concerning “Vaccine” and “Covid” related tweets are shown in Figs. 10, 11, 12 and 13.

Fig. 11 Sentiment of tweets about vaccine in December 2020

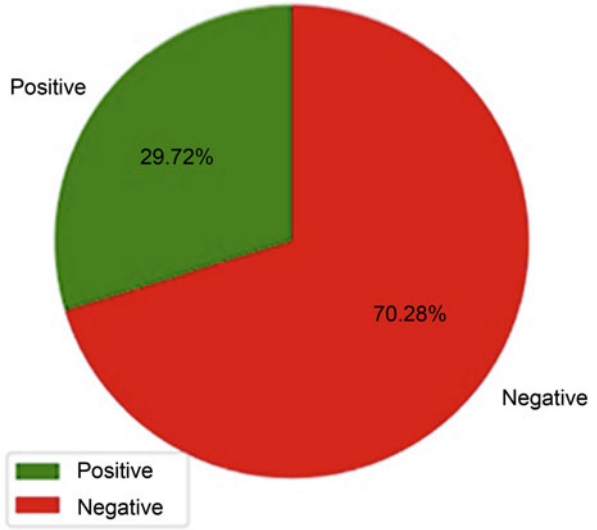


Fig. 12 Sentiment of tweets about Covid in January 2021

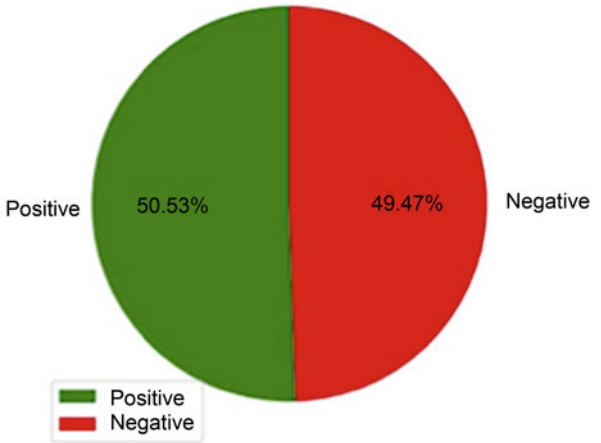
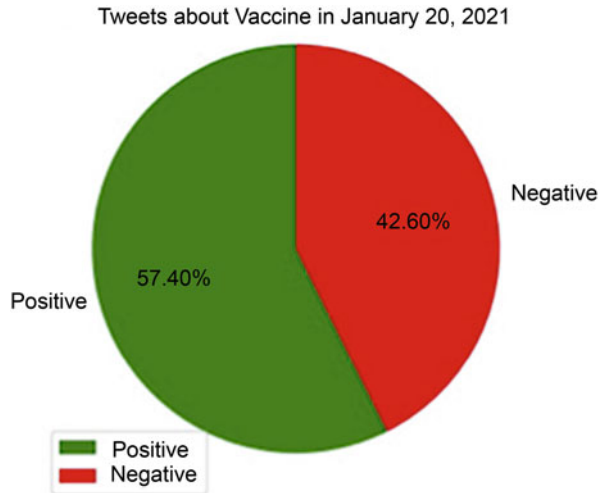


Fig. 13 Sentiment of tweets about vaccine in January 2021



5 Conclusion

As a result of this study, several conclusions could be derived. First of all, for the sentiment analysis algorithm, 0.6 accuracy was determined. This accuracy can be developed with further methods of preprocessing or with a better and much more efficient training algorithm. Also, the algorithm include just positive and negative evaluation. This can be expanded through more complex and a better algorithm with adding the neutrality. Even further, some evaluation techniques can be used with different degrees. All these evaluations are effective in our results. We can see the most occurred words in the tables and changes through the months in that trend. Also we see that negativity is seen more in the Covid tweets, whereas positivity is seen more in Vaccine tweets. But this result can be doubted since accuracy is 0.6 and also algorithm omits the neutral tweets. These results should be considered for further developments and works.

References

1. Chawan P (2012) Sentiment analysis and influence tracking using Twitter. *Int J Adv Res Comput Sci Elect Eng*, 1
2. Cossu J, Dugué N, Labatut V (2015) Detecting real-world influence through Twitter. In: 2015 second European network intelligence conference, Karlskrona, pp 83–90
3. Sanandres E, Llanos R, Camilo MO (2018) Topic modeling of Twitter conversations
4. Doan S, Yang EW, Tilak SS et al (2019) Extracting health-related causality from twitter messages using natural language processing. *BMC Med Inform Decis Mak* 19, 79

5. Alowisheq A, Alrajebah N, Alrumikhani A, Al-Shamrani G, Shaabi M, Al-Nufaisi M, Alnasser A, Al-Humoud S (2017) Investigating the relationship between trust and sentiment agreement in Arab Twitter users, pp 236–245
6. Tago K, Jin Q (2018) Influence analysis of emotional behaviors and user relationships based on Twitter data. *Tsinghua Sci Technol* 23(1):104–113. <https://doi.org/10.26599/TST.2018.9010012>
7. Li P, Zhao W, Yang J, Wu J (2019) CoTrRank: trust evaluation of users and tweets. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence. Twenty-eighth international joint conference on artificial intelligence IJCAI-19
8. Bae Y, Lee H (2012) Sentiment analysis of twitter audiences: measuring the positive or negative influence of popular twitterers. *J Am Soc Inf Sci Technol* 63(12):2521–2535
9. Cano Basave AE, Mazumdar S, Ciravegna F (2014) Social influence analysis in microblogging platforms – a topic-sensitive based approach. *Semantic Web* 5(5):357–403
10. Sung J, Moon S, Lee J-G (2013) The influence in Twitter: are they really influenced? In: Behavior and social computing. Springer International Publishing, New York City, pp. 95–105
11. Hong L, Davison BD (2010) Empirical study of topic modeling in Twitter. In: Proceedings of the first workshop on social media analytics – SOMA '10. The First Workshop
12. Grant C, George C, Jenneisch C, Wilson J (2011) Online topic modeling for real-time Twitter search, NIST Special Publication: SP 500-296, The Twentieth Text REtrieval Conference (TREC 2011) Proceedings. <https://trec.nist.gov/pubs/trec20/t20.proceedings.html> (accessed July 1, 2022)
13. Jonsson E (2016) An evaluation of topic modelling techniques for Twitter. <http://www.cs.toronto.edu/~jstolee/projects/topic.pdf> (accessed July 1, 2022)
14. Ruan Y, Durrezi A, Alfantoukh L (2018) Using Twitter trust network for stock market analysis. *Knowl Based Syst* 145:207–218