



Semantic Fake News Detection: A Machine Learning Perspective

Adrian M. P. Braşoveanu^{1,2(✉)} and Răzvan Andonie³

¹ Electronics and Computers Department, Transilvania University of Braşov,
Braşov, Romania

² MODUL Technology GmbH, Vienna, Austria
adrian.brasoveanu@modul.ac.at

³ Computer Science Department, Central Washington University,
Ellensburg, WA, USA
andonie@cwu.edu

Abstract. Fake news detection is a difficult problem due to the nuances of language. Understanding the reasoning behind certain fake items implies inferring a lot of details about the various actors involved. We believe that the solution to this problem should be a hybrid one, combining machine learning, semantics and natural language processing. We introduce a new semantic fake news detection method built around relational features like sentiment, entities or facts extracted directly from text. Our experiments show that by adding semantic features the accuracy of fake news classification improves significantly.

Keywords: NLP · Semantics · Relation Extraction · Deep learning

1 Introduction

Teaching an automated system to recognize fake news is a challenging task, especially due to its interdisciplinary nature. At a superficial level it is important to distinguish between satire and political weapons (or any other kind of weapons built on top of deceptive news) [4], but when examining a news item, it might help to deploy a varied Natural Language Processing (NLP) arsenal that includes sentiment analysis, Named Entity Recognition Linking and Classification (NERLC [12]), n-grams, topic detection, part-of-speech (POS) taggers, query expansion or relation extraction [34]. Quite often such tools are supported by large Knowledge Bases (KBs) like DBpedia [16], which collects data about entities and concepts extracted from Wikipedia. The extracted named entities and relations will be linked to such KBs whenever possible, whereas various sentiment aspects, polarity or subjectivity might be computed according to the detected entities. Features like sentiment, named entities or relations render a set of shallow meaning representations, and are typically called *semantic features*. In contrast, POS or dependency trees render *syntactic features*.

The underlying assumption made by most models used for detecting fake news is that the title and style of an article are sufficient to identify it as fake news. This is mostly true for news that originate from verifiable bad sources, which is rarely the case anymore. Therefore, we think that taking a holistic approach, that includes a machine generated Knowledge Graph (KG) [20] of all the stakeholders involved in the various events we are interested in is absolutely needed. Such a holistic approach includes methods which can generate and learn graphs of entities associated to fake news.

Our contribution is a method used to integrate semantic features in the training of fake news classifiers. The goal is to show how to use semantic features to improve fake news detection. For this, we compute semantic features (sentiment analysis, named entities and relations) which will be added to a set of syntactic features (POS - part-of-speech and NPs - Noun Phrases) and to the features of the original input dataset. On the resulted augmented dataset we apply various classifiers, including Deep Learning (DL) models: Long-Short Term Memory (LSTM), Convolutional Neural Network (CNN), and Capsule Networks. For the Liar data set [32], using semantic features improves the fake news recognition accuracy on average by 5–10%.

The paper is organized as follows. Section 2 presents the most recent results in fake news recognition. Section 3 introduces our approach for building machine generated KGs for semantic fake news detection. Section 4 describes the experimental results. The paper is concluded in Sect. 5.

2 Related Work

An exploration of the fake news phenomena during more than a decade (2006–2017) was built around Twitter rumor cascade by a series of social scientists [31]. Multiple surveys (e.g., [26, 35]) were focused on building various fake news classifications. Rubin [22] defined a set of criteria for creating a good text corpora for fake news detection, namely that (i) it should only contain verifiable facts, (ii) happened in a certain interval, and (iii) were reported using similar style though with various degrees of cultural influences. Any such corpora should only focus on text-only item, as they would be easier to process.

Most of the time, simply analyzing the text will not get us very far. Recent models incorporate some data about the networks (e.g., social media, organizations) through which the news was spread. Ruchansky [23] proposed a CSI model which stands for Capture, Score and Integrate, therefore combining information on the temporal activity of the users, their behavior, and a classifier. The 3HAN network [28] is a Hierarchical Attention Network (HAN) network with three layers used to examine different parts of articles.

A model for early detection of fake news based on news propagation paths is described in [17] and is based on a hybrid time-series classifier that contains both Recurrent Neural Networks (RNNs) and CNNs. Wu [33] assumed that intentional fake news are typically manipulated to look like real news. He built a classifier based on social media propagation pathways using LSTM-RNN and

embeddings. Vo and Lee [30] took a different approach, focusing on the story told by fake news URLs and the co-occurrence of various entities through such links.

A set of LSTMs was used for performing a multi-source multi-class fake news detection (or MMFD) in [14]. The advantage of this method is the multi-source fusion of the MMFD framework, since it can determine various degrees of fake news. The accuracy of the approach is not very high, but given the fact that it combines three large components (automated feature extraction, multi-source fusion and fakeness discrimination) it is promising. Aghakhani [2] showed that a Generative Adversarial Network (GAN) [8] can perform relatively well for detecting deceptive reviews.

A good review of the state-of-the-art DL applications in NLP, that also includes details about sentiment analysis or named entities extraction/classification, is [34].

3 Our Approach

In this section, we introduce our approach for semantic information extraction and then describe how we use the extracted information to classify fake news. We present techniques related to: metadata collection, extraction of relations, and inclusion of embeddings to neural classifiers.

Our main research question is: *what are the most useful semantic features for improving fake news detection?* Ideally, such features should be integrated into the neural models, whenever possible. Today, due to the cost of developing good semantic systems, some of these features might come from various external tools. The semantic features need to be selected according to the task and dataset at hand. If the task refers to the detection of fake news as spread by people via their statements, then the main entities we will be interested in might include people, organizations, locations and events.

In order to fully exploit the relations between the entities mentioned in a news statement, our procedure includes the following steps:

- **Metadata collection.** The first step is to simply collect the sentiment, entities and additional metadata available from third party tools.
- **Relation Extraction.** A second pass will collect both (i) the general relations found in a KG, and (ii) those computed from the current texts.
- **Embeddings.** Last step refers to the adaptation of various neural models (e.g., by adding a layer of embeddings) for improving fake news detection.

The features included in the last step will be only internal, whereas the features included on the other steps can also be external. The entire process is illustrated in Fig. 1.

The intuition behind the current data modeling that lead to the additional semantic features is that by adding extracted entities and making a clear distinction between direct and indirect speech, we can create the premises for more sophisticated analysis that may pinpoint the personal history of a speaker with

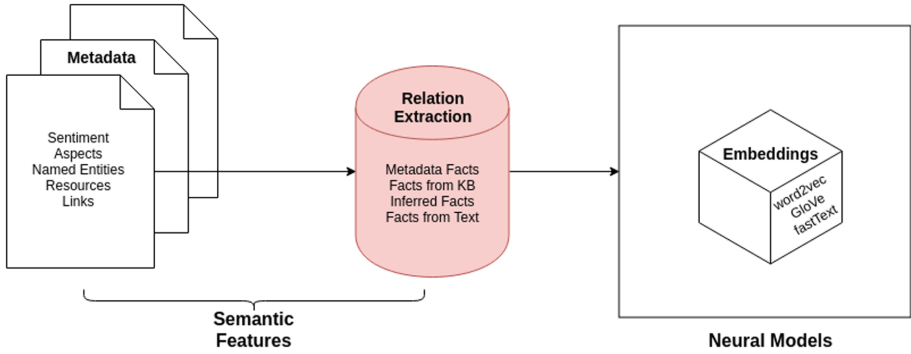


Fig. 1. External and internal semantic features for neural network models.

both the issue at hand (or subject), as well as with all the parties involved in the respective issue. If such an analysis is extended, down the road, it should also be possible to identify more obscure details about a speaker, for example if (s)he follows the party line or not. In other words, it opens up the possibility of using the graphs to peak behind the scenes of various declarations.

3.1 Fake News Detection and Knowledge Graphs

There are various definitions of fake news. Most of them refer to Alcott and Gentzkow’s paper that examines the impact of fake news on the 2016 US Election [3].

Definition (based on [3]). *A news item or a part of a news item will be considered fake if it can be verified that its content is false.*

In order to perform semantic fake news detection, some additional statements like the past truth history of a speaker or the relations between speakers and publishers should be considered if possible. The idea of using past inaccuracies for each speaker was introduced with the Liar data set [32] and named credit history, but it is rarely used in practice.

Definition (based on [32]). *Credit History (CH) is the historical count of false (or provably untrue) statements for an actor.*

A credit history score can also be replaced by a single aggregated count of all the untrue values. Such credit scores allows us to understand diverse perspectives when analyzing news and helps determine which person or group might benefit from spreading certain news. An earlier iteration of this idea was also explored in the context of social media networks: credibility propagation [13].

Definition. *A credit history graph is a graph that contains all the entities, their credit histories and links between them as they are available from a Knowledge Graph (KG) or generated from a collection of texts.*

Relational features can be considered an alternative to the credit history features and can be extracted from both traditional KGs (e.g., DBpedia, Wikidata), as well as from text.

Definition. *Relational features include all the features extracted directly from the texts or the named entities detected in them through the exploitation of Knowledge Graphs.*

While we focus here on extracting all the needed features directly from the data at hand (the text), the Tri-Relationship framework described in Shu’s paper [27] also deserves a mention here, even though it is focused on the objects involved in distributing the news (e.g., people, organizations). All the mentioned approaches share the idea of enriching the fake news text with a set of annotations, in order to provide some context.

3.2 Metadata Collection Pipeline

Our pipeline for generating metadata has following components:

- *Sentiment Analysis (SA)*. Sentiment annotations can exist on multiple levels: (i) document; (ii) sentence; (iii) aspect-based [34]. Current state-of-the-art systems are typically aspect-based, therefore all the aspect of the entity features can get an estimate of the sentiment value. Since our data set (the Liar data) contains short statements, we use aggregated sentence level sentiment polarity and subjectivity values.
- *Named Entities (NE)*. Since the results for NE extractions are typically good enough [12], almost any modern NLP library can be used for this task.
- *Named Entity Links (NEL)*. Generally NERLC (NER+linking and classification) tasks are considered more complicated and typically require dedicated NEL engines [12]. Any good NEL engine can be used for this task. We use a wrapper built on top of DBpedia Spotlight [7].

3.3 Relation Extraction

Instead of using existing solutions, we develop a simple *Relation Extraction (REL)* component that queries DBpedia. Where possible, the existing entities are enriched with additional data obtained via a SPARQL query from DBpedia. This is particularly important in order to discover more relations between a speaker (which we will call *source entity*) and his/her subject (which we will call *target entity*). We consider two types of relations:

- (i) extracted directly from the provided news statements by defining the types of relations we are interested in via POS tags (for example, for extracting relations between two entities we will generally be interested in NP - V - NP chains - a verb between two proper nouns, whereas additional relations for an entity can be added by extracting S - V - O triplets);

- (ii) extracted from the DBpedia Knowledge Base (e.g., if `dbr:Donald.Trump` mentions `dbr:Barack.Obama` in a document, all the triples that belong to these entities are extracted from DBpedia and a subset of common links like `dbo:orderInOffice` or `dbo:President` is identified).

The machine generated KG includes all the DBpedia triples that belong to the entities collected from the data set. The relations extracted from text are schemaless, whereas the relations extracted from KG are grounded to a schema (e.g., DBpedia ontology). This component is implemented with the Python libraries RDFLib, SPARQLWrapper and Spacy¹.

3.4 Embeddings

Shallow neural architectures that learn word embeddings from distributional semantics (e.g., continuous bag of words architectures like Word2Vec, GloVe or fastText [19]) have been successfully applied to classic NLP problems [34], and should be an integral part of any NLP architecture. Such architectures generally provide fast computation times and lead to good results due to the fact that they capture relational similarities.

If the used corpora is clean and large enough (several tens of thousands of examples [19]), embeddings can be an ideal solution for building baselines. Only the most used (word2vec, GloVe, fastText) pre-computed embeddings were included for the top 60k English words. The component that loads them uses negative sampling and a fixed size of 300. The Keras API offers the possibility to add an embeddings layer to a neural network. This layer can be used for: (i) learning and saving the embeddings together with the word vectors; (ii) loading pre-trained embeddings. In all our DL models, we place such a layer after the inputs and use it for loading embeddings. Such a layer is effective especially when the number of training examples is relatively small [21].

4 Experiments

The success of our approach depends on a series of components for extracting sentiment scores, named entities, or relations. Therefore, if those components do not perform well, the whole approach will be flawed. First, we would like to find out if such an approach is valid. Therefore, missing a named entity from a statement might not be extremely important at this stage. If the approach proves to be valid, then further work needs to include additional evaluations for all the components in the pipeline, or at least some of their performance scores (when available).

We use the Liar data set [32] for our experiments. It contains politics-related articles classified based on the degree of truth, while also offering credit histories that tracks the accuracy of the speaker statements. The data set is split into three partitions (**train**, **test** and **validation**) and includes six classes that need

¹ <https://spacy.io/>.

to be predicted: **False**, **Barely-true**, **Half-true**, **Mostly-true**, **True**, **Pants on fire**. The initial paper about the Liar data set [32] identified SVMs as best classical models and CNNs as the best Deep Learning classifiers. A follow-up paper [18] indicates that LSTMs would be even better. Since our focus is not on credit history (five counts for all the classes that are not True including the score for the current statement) but on the impact of the relational features, we do not reproduce those results and do not compare with them.

Table 1. Accuracy for the test set runs on the Liar dataset. The best results are presented in **bold**. **T** stands for text, **A** for attributes and **R** for relations.

Model	T	T+A	T+R	ALL
<i>Classic ML</i>				
Multinomial Naive Bayes	0.230	0.234	0.244	0.269
SGDClassifier	0.229	0.216	0.225	0.246
Logistic regression (OneVsRest)	0.226	0.240	0.249	0.262
Random forest	0.249	0.219	0.231	0.238
Decision trees	0.226	0.240	0.249	0.262
SVM	0.240	0.250	0.259	0.284
<i>Deep Learning</i>				
CNN	0.260	0.274	0.280	0.290
BasicLSTM	0.225	0.255	0.265	0.324
BiLSTM attention	0.408	0.435	0.444	0.496
GRU attention	0.460	0.496	0.506	0.549
CapsNet	0.555	0.583	0.593	0.644

We consider four cases, as depicted in Table 1. The texts themselves (named **text (T)**) are simply statements that are taken out of their original context. The features included in the original data set (**text+attributes (T+A)**) contain information about the subject, speaker (including his job title, state and party affiliation), as well as credit history, and the context (the speech’s location). The set **text+relations (T+R)** has semantic features (sentiment polarity, sentiment subjectivity, entities, links, and relations), syntactic features (NP), and the aggregated score of the credit history counts. The features included in the **T+R** data set are all extracted directly from the statements - there is no need to use the full text of the articles to compute them. This is an important detail, since this operation can always be performed if we have a good set of tools for metadata generation, even when the full articles are no available. The last set of features (identified as **all (ALL)**) includes all the previous features.

The classes are balanced and the split between train and test is 4:1. In Tables 1 and 2 we report the test set accuracy scores for all considered models and additional features.

Table 2. Accuracy for the test set runs using different combinations of semantic profile attributes (**T+R**). The best results are presented in **bold**.

Feature	Acc
BasicLSTM (T+A)	0.255
Sentiment Polarity	0.260
Sentiment Subjectivity	0.267
NP	0.273
Entities	0.284
Links	0.285
Relations	0.303
CH	0.324
All	0.324

We start by testing several “classic” models [10] that were built with scikit-learn (Table 1). For these models, using the relational features (**T+R**) shows some improvements, typically 2–3% above the original features (**T+A**) of the data set. However, the best score are far from optimal. Logistic regression and decision trees scores prove to be quite similar for all the three runs, while simultaneously being the worst scores. We notice a single case (the random forest classifier) in which the added relational features do not yield improvements over a run with only the original text. The best “classic” ML classifier proves to be the SVM, confirming the results from [18].

In the second phase, we test several DL models. The DL models are built with Keras [6] and TensorFlow [1], and use hot encoding of the class labels. For the DL models, the reported evaluation metric is accuracy with Adam optimizer [15].

The following DL classifiers are used:

- CNN - based on the model described in [18].
- BasicLSTM - a simple LSTM with a GlobalMaxPool layer, dropout set at 0.1 and dense layers;
- BiLSTM [5] - a bidirectional LSTM with attention, dropout and recurring dropout set at 0.1, which also includes an embeddings layer and the rest of the layers from the BasicLSTM;
- GRU [11] - a GRU with attention, otherwise similar to the previous BiLSTM model;
- CapsNetLSTM [24] - uses a Capsule layer instead of the GlobalMaxPool layer used in the other models.

All the DL models, besides CNN and BasicLSTM, use embeddings. We did not perform additional tuning of the DL models. We noticed that the embeddings for the most used 60k words from the English language have almost no effect on the results. The input vectors were loaded using Keras’s embeddings layers which

is defined as the first hidden layer of a network. For the DL experiments, we used whenever possible pre-trained models. Of course, fine-tuning the architectures may improve these results.

In all cases, relational features (**T+R**) perform better than the original features of the data set (**T+A**), which suggests that in some cases it might be enough to simply collect texts and build the rest of the features from metadata.

We note that all the DL models obtain better scores than the classic models with the same features. While the current literature is mostly focused on CNNs and basic LSTMs, we observe that attention models and CapsNet models performed best. For all DL models, adding our features results in an accuracy increase of up to 5–6%. This could be caused by the fact that the embeddings represent internal features of our DL models.

We have not repeated all feature combinations presented in Wang [32] and Long [18], but rather took the best feature combinations found in those papers and added new combinations based on the relational features proposed by us. The scores obtained by us for SVMs, basic CNNs and LSTMs confirm their results. Using relational features (sentiment, recognized named entities, named entities links, relations) together with syntactic features (NP), it is already possible to beat the baselines at a comfortable distance, even without using advanced architectures. It is even possible to use only these semantic and syntactic features, instead of the original ones, and the scores will still be better than the baselines.

We tried to minimize the number of input features. Depending on the length of the text and number of entities involved, the number of additional features can be increased - which may lead to some increase in the overall performance. The most important thing when using our technique is to select the appropriate additional features that can lead to performance improvements. According to the results (Table 2), a good choice is to select relations, sentiments and entities.

5 Conclusions

While the literature on fake news detection is increasing at fast pace, the accuracy of the various models greatly varies depending on the data sets and the number of classes involved. In our view, good models should be adaptive and should not require a lot of fine-tuning on data sets. According to our results, by also considering relational features like sentiment, named entities or facts extracted from both structured (e.g., Knowledge Graphs) and unstructured data (e.g., text), we generally obtain better scores on most classifiers.

Currently, most models are based on word embeddings, even though phrases and multi-words expressions perform better for longer texts. This is due to the fact that the language used in a fake news article may differ from the language used in a normal article, as it is often needed to reinforce certain claims. Some future investigation areas include exploiting these relational features together with graph neural networks, like the recently developed R-GCN [25] or using a single multi-head attention architecture [29] to generate all the semantic features. Another interesting direction is to use semantic features for detecting fake

reviews. While this is somewhat similar to the fake news detection, the goal here is to detect fake accounts on websites like TripAdvisor or fake authorships.

References

1. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. CoRR abs/1605.08695 (2016). <http://arxiv.org/abs/1605.08695>
2. Aghakhani, H., Machiry, A., Nilizadeh, S., Kruegel, C., Vigna, G.: Detecting deceptive reviews using generative adversarial networks. CoRR abs/1805.10364 (2018). <http://arxiv.org/abs/1805.10364>
3. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. *J. Econ. Perspect.* **31**(2), 211–236 (2017)
4. Berghel, H.: Lies, damn lies, and fake news. *IEEE Comput.* **50**(2), 80–85 (2017). <https://doi.org/10.1109/MC.2017.56>
5. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. *TACL* **4**, 357–370 (2016). <https://transacl.org/ojs/index.php/tacl/article/view/792>
6. Chollet, F.: *Deep Learning with Python*. Manning Publications Co., Shelter Island (2017)
7. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Sabou, M., Blomqvist, E., Noia, T.D., Sack, H., Pellegrini, T. (eds.) *I-SEMANTICS 2013–9th International Conference on Semantic Systems, ISEM 2013, Graz, Austria, 4–6 September 2013*, pp. 121–124. ACM (2013). <https://doi.org/10.1145/2506182.2506198>
8. Goodfellow, I.J., et al.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, 8–13 December 2014, Montreal, Quebec, Canada*, pp. 2672–2680 (2014). <http://papers.nips.cc/paper/5423-generative-adversarial-nets>
9. Guyon, I., et al. (eds.): *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA* (2017)
10. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. SSS. Springer, New York (2009). <https://doi.org/10.1007/978-0-387-84858-7>. <http://www.worldcat.org/oclc/300478243>
11. Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., Ney, H.: LSTM, GRU, highway and a bit of attention: an empirical overview for language modeling in speech recognition. In: Morgan, N. (ed.) *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, 8–12 September 2016*, pp. 3519–3523. ISCA (2016). <https://doi.org/10.21437/Interspeech.2016-491>
12. Ji, H., Nothman, J.: Overview of TAC-KBP2016 tri-lingual EDL and its impact on end-to-end KBP. In: *Eighth Text Analysis Conference (TAC)*. NIST (2016). <https://tac.nist.gov/publications/2016/additional.papers/>
13. Jin, Z., Cao, J., Zhang, Y., Luo, J.: News verification by exploiting conflicting social viewpoints in microblogs. In: Schuurmans, D., Wellman, M.P. (eds.) *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 12–17 February 2016, Phoenix, Arizona, USA*, pp. 2972–2978. AAAI Press (2016). <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12128>

14. Karimi, H., Roy, P., Saba-Sadiya, S., Tang, J.: Multi-source multi-class fake news detection. In: Bender, E.M., Derczynski, L., Isabelle, P. (eds.) Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, 20–26 August 2018, pp. 1546–1557. Association for Computational Linguistics (2018). <https://aclanthology.info/papers/C18-1131/c18-1131>
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014). <http://arxiv.org/abs/1412.6980>
16. Lehmann, J., et al.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web* 6(2), 167–195 (2015). <https://doi.org/10.3233/SW-140134>
17. Liu, Y., Wu, Y.B.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, 2–7 February 2018. AAAI Press (2018). <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16826>
18. Long, Y., Lu, Q., Xiang, R., Li, M., Huang, C.: Fake news detection through multi-perspective speaker profiles. In: Kondrak, G., Watanabe, T. (eds.) Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, 27 November–1 December 2017, Volume 2: Short Papers, pp. 252–256. Asian Federation of Natural Language Processing (2017). <https://aclanthology.info/papers/I17-2043/i17-2043>
19. Mikolov, T., Grave, E., Bojanowski, P., Puhusch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Calzolari, N., et al. (eds.) Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, 7–12 May 2018. European Language Resources Association (ELRA) (2018). <http://www.lrec-conf.org/lrec2018>
20. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104(1), 11–33 (2016). <https://doi.org/10.1109/JPROC.2015.2483592>
21. Qi, Y., Sachan, D.S., Felix, M., Padmanabhan, S., Neubig, G.: When and why are pre-trained word embeddings useful for neural machine translation? In: Walker, M.A., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, 1–6 June 2018, Volume 2 (Short Papers), pp. 529–535. Association for Computational Linguistics (2018). <https://aclanthology.info/papers/N18-2084/n18-2084>
22. Rubin, V., Conroy, N., Chen, Y., Cornwell, S.: Fake news or truth? using satirical cues to detect potentially misleading news. In: Proceedings of the Second Workshop on Computational Approaches to Deception Detection, pp. 7–17 (2016)
23. Ruchansky, N., Seo, S., Liu, Y.: CSI: a hybrid deep model for fake news detection. In: Lim, E., et al. (eds.) Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, 06–10 November 2017, pp. 797–806. ACM (2017). <https://doi.org/10.1145/3132847.3132877>
24. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Guyon et al. [9], pp. 3859–3869. <http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules>
25. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38

26. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: a data mining perspective. *SIGKDD Explor.* **19**(1), 22–36 (2017). <https://doi.org/10.1145/3137597.3137600>
27. Shu, K., Wang, S., Liu, H.: Exploiting tri-relationship for fake news detection. *CoRR* abs/1712.07709 (2017). <http://arxiv.org/abs/1712.07709>
28. Singhania, S., Fernandez, N., Rao, S.: 3HAN: a deep neural network for fake news detection. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S. (eds.) *ICONIP 2017*. LNCS, vol. 10635, pp. 572–581. Springer, Cham (2017)
29. Vaswani, A., et al.: Attention is all you need. In: Guyon et al. [9], pp. 6000–6010. <http://papers.nips.cc/paper/7181-attention-is-all-you-need>
30. Vo, N., Lee, K.: The rise of guardians: fact-checking url recommendation to combat fake news. In: Collins-Thompson, K., Mei, Q., Davison, B.D., Liu, Y., Yilmaz, E. (eds.) *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, 08–12 July 2018*, pp. 275–284. ACM (2018). <https://doi.org/10.1145/3209978.3210037>
31. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. *Science* **359**(6380), 1146–1151 (2018)
32. Wang, W.Y.: Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection. *CoRR* abs/1705.00648 (2017). <http://arxiv.org/abs/1705.00648>
33. Wu, L., Liu, H.: Tracing fake-news footprints: characterizing social media messages by how they propagate. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (eds.) *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, 5–9 February 2018*, pp. 637–645. ACM (2018). <https://doi.org/10.1145/3159652.3159677>
34. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing [review article]. *IEEE Comp. Int. Mag.* **13**(3), 55–75 (2018). <https://doi.org/10.1109/MCI.2018.2840738>
35. Zannettou, S., Sirivianos, M., Blackburn, J., Kourtellis, N.: The web of false information: rumors, fake news, hoaxes, clickbait, and various other shenanigans. *CoRR* abs/1804.03461 (2018). <http://arxiv.org/abs/1804.03461>