Information Systems Frontiers (2018) 20:925–932
https://doi.org/10.1007/s10796-018-9847-6

# An Embedding Based IR Model for Disaster Situations

Ayan Bandyopadhyay[1] · Debasis Ganguly[2] · Mandar Mitra[1] · Sanjoy Kumar Saha[3] · Gareth J.F. Jones[4]

Published online: 4 April 2018

## Abstract

Twitter (http://twitter.com) is one of the most popular social networking platforms. Twitter users can easily broadcast disaster-specific information, which, if effectively mined, can assist in relief operations. However, the brevity and informal nature of tweets pose a challenge to Information Retrieval (IR) researchers. In this paper, we successfully use word embedding techniques to improve ranking for ad-hoc queries on microblog data. Our experiments with the 'Social Media for Emergency Relief and Preparedness' (SMERP) dataset provided at an ECIR 2017 workshop show that these techniques outperform conventional term-matching based IR models. In addition, we show that, for the SMERP task, our word embedding based method is more effective if the embeddings are generated from the disaster specific SMERP data, than when they are trained on the large social media collection provided for the TREC (http://trec.nist.gov/) 2011 Microblog track dataset.

**Keywords** Microblog · Twitter · Information retrieval · Word embedding

## 1 Introduction

Social media have become very important sources of information in disaster situations (Imran et al. 2015; Varga and et al. 2013). Social media provide information to both victims of the disaster, as well as providers of relief and resources. Information Retrieval (IR) techniques can provide a means for effectively and efficiently sharing this information both within and between these two groups. Recognizing the importance of developing robust IR tools for locating relevant content in social media sources, IR benchmark tasks, specifically the FIRE 2016 Microblog track (Ghosh and Ghosh 2016) and SMERP 2017 (Ghosh et al. 2017) tasks, have been organized to provide evaluation exercises for testing such tools and techniques.

Traditional approaches to IR typically rely on word-matching based retrieval methods like BM25 (Robertson et al. 1994) and Language Modeling (LM) (Hiemstra 2000; Ponte and Croft 1998). These approaches perform well only when queries and documents use the same vocabulary, and both are sufficiently detailed. However, such models are likely to perform poorly if relevant documents lack the important terms present in the query. This problem can be particularly acute for microblog posts,[1] since their brevity (at most 140 characters) means that they may not contain many of the necessary keywords. Consequently, tweets retrieved from a tweet collection using these models may not contain relevant documents at top ranks. The widespread occurrence of spelling variations, abbreviations and code-mixing in social media generally makes IR from tweets even more difficult.

The vocabulary mismatch problem may be alleviated by the use of semantic matching techniques which involve

✉ Ayan Bandyopadhyay
bandyopadhyay.ayan@gmail.com

Debasis Ganguly
debasis.ganguly1@ie.ibm.com

Mandar Mitra
mandar@isical.ac.in

Sanjoy Kumar Saha
sks_ju@yahoo.co.in

Gareth J.F. Jones
Gareth.Jones@dcu.ie

[1] Indian Statistical Institute, Kolkata, India

[2] IBM Research, Dublin, Ireland

[3] Jadavpur University, Kolkata, India

[4] Dublin City University, Dublin, Ireland

[1] hereafter referred to as tweets

matching the semantic intent of the query with the relevant documents, without explicitly needing the presence of all the query key terms in the document. With this hypothesis in mind, we investigate the use of word embeddings for tweet retrieval in this paper. We use word embeddings generated by the popular Word2vec (Mikolov et al. 2013) tool, and compare conventional word matching models (LM and BM25) with document embedding based retrieval. We find that the latter produces better results than the former, mainly because the semantic matching techniques are able to bridge vocabulary mismatch between queries and documents. This finding is consistent with those of earlier studies that explored the use of word embedding methods on non-disaster datasets (Diaz et al. 2016).

The remainder of the paper is organized as follows. We discuss related work in Section 2. We then introduce our proposed approach

in Section 3. Experimental results are presented in Section 4. We conclude by sketching some directions for further work in Section 5.

## 2 Related Work

Early work on microblog retrieval includes a language modeling approach for searching Microblog posts proposed by Massoudi et al. (2011). Their method incorporated query expansion and used certain "quality indicators" during matching. Hashtag retrieval (Efron 2010) is also closely related to our work. Hashtags[2] refer to certain important "keywords" in a message that are designated as tags using a hash (#) sign. Hashtags are useful as a very quick method for categorizing or tagging messages. Efron et al. (2010) showed that for a Twitter collection, hashtags can be predicted using query expansion. Dong et al. (2010) proposed a ranking method that takes both "relevance" and freshness into account. Del Corso et al. (2005) also suggested a ranking method for news documents in which recency plays a major part. Bandyopadhyay et al. (2012) studied the use of external resources for query expansion during tweet retrieval.

Word embeddings map words in a text collection to relatively low-dimensional real-valued vectors. These vectors are supposed to capture the semantics (or more precisely the contexts or usage patterns) of words within the text. Word2vec (Mikolov et al. 2013a, b) proposed by Mikolov et al. is an efficent word embedding method that has received much attention in the last few years. We use this method in our proposed model. This idea was later extended in (Le and Mikolov 2014) to sentences and documents. Lau et al. (2016) present an empirical evaluation of these ideas, along with some practical insights into document embedding generation. Ganesh et al. (2016) describe a two-phase document embedding method. Recently, Kim et al. (2017) proposed the use of word embeddings to represent a documents' concepts. They create clusters of semantically similar words from documents, following which a document is represented as a bag-of-concepts, rather than the traditional bag-of-words. Xing et al. (2014) use word embeddings within a document classification method. They represent a document as a Gaussian Mixture Model estimated from the constituent word vectors.

## 3 Our Approach

As mentioned in the Introduction, our objective is to study whether word embeddings may be used to improve retrieval effectiveness over traditional IR models.

**Proposed Model** We propose the use of *document embeddings* to represent each document as a vector. Our model is a straightforward extension of word2vec (Mikolov et al. 2013b) to the document level. Suppose document $d$ contains a set of words $W_d$, then the vector representation of $d$ is given by

$$\overrightarrow{d} = \sum_{i=0}^{|W_d|} \overrightarrow{w_{id}} \tag{1}$$

where $w_{id} \in W_d$ is the $i^{th}$ distinct word of document $d$, and $\overrightarrow{w_{id}}$ is the word2vec embedding of $w_{id}$. The same representation is used for each query, say $q$, which can similarly be viewed as a set of words. Let $\overrightarrow{q}$ be the embedding vector of the query $q$. Then the similarity between $q$ and $d$ may be calculated by the following scoring function:

$$Sim(q,d) = CosSim(\overrightarrow{q}, \overrightarrow{d}) = \frac{\overrightarrow{q} \cdot \overrightarrow{d}}{||\overrightarrow{q}|| \cdot ||\overrightarrow{d}||} \tag{2}$$

where $CosSim$ is the *cosine similarity* between the vectors, and $\overrightarrow{q} \cdot \overrightarrow{d}$ represents the inner-product between $\overrightarrow{q}$ and $\overrightarrow{d}$.

For the word embeddings used above, we tried both the models that are commonly used with word2vec:

- $V1$: 'Continuous bag of words' (Mikolov et al. 2013a), and
- $V2$: 'Skip gram' (Mikolov et al. 2013b).

[2] https://support.twitter.com/entries/49309-what-are-hashtags-symbols

### 3.1 Baselines

Next, we briefly review the baseline methods against which our proposed model is compared.

#### 3.1.1 Language Model

The general idea of language modeling based retrieval can be described in the following way. Let $Q$ be a query and $d$ be a document. Let $\mathcal{D}$ represent the language model estimated from $d$. Then the score of document $d$ with respect to query $Q$ is given by $p(Q|\mathcal{D})$, the probability of generating $Q$ from $\mathcal{D}$. The language model $\mathcal{D}$ associated with the document $d$ is usually approximated by a unigram language model, i.e., a probability distribution over single words. Assuming that any pair of distinct terms occurs independently, the retrieval score of $d$ for a given query $Q$ can be written as

$$\begin{aligned} \text{Score}(Q, d) &= p(Q|\mathcal{D}) \\ &= \prod_{q \in Q} p(q|d) \end{aligned} \tag{3}$$

#### 3.1.2 Jelinek-Mercer Smoothing (LM-JM)

(Jelinek and Mercer 1980)

To overcome the zero probability problem (when a query term is missing from document $d$, its score becomes zero according to Eq. 3), the language model $\mathcal{D}$ is smoothed by interpolating the Maximum Likelihood Estimate (MLE) of $p(w_i|d)$ with a background language model, estimated from the entire collection $C$, as in Eq. 4.

$$\begin{aligned} p(Q|\mathcal{D}) &= \prod_{q \in Q} [(1-\lambda)p(q|d) + \lambda p(q|C)] \\ &= \prod_{q \in Q} \left[ (1-\lambda)\frac{tf(q,d)}{|d|} + \lambda \frac{cf(q)}{|C|} \right] \end{aligned} \tag{4}$$

Here, $tf(q,d)$ and $cf(q)$ indicate the number of occurrences of $q$ in the document $d$, and in the whole document collection $C$, respectively; $|d|$ and $|C|$ represent the total number of words in the document and the collection respectively; $\lambda = [0, 1]$ is the interpolation parameter. This language modeling based retrieval model is known as language model with Jelinek-Mercer smoothing or linear smoothing, and is referred to as JM or, LM-JM in the rest of this article.

#### 3.1.3 Dirichlet Smoothing (LM-D)

(MacKay and Peto 1994)

Another smoothing method that is also used by researchers is the Dirichlet prior smoothing method. The mathematical form of this model is similar to LM-JM (Eq. 4), except that the interpolation parameter is a function of document length instead of being a constant (see Eq. 5).

$$P(Q|\mathcal{D}) = \prod_{q \in Q} \frac{tf(q,d) + \mu p(q|C)}{|d| + \mu} \tag{5}$$

#### 3.1.4 Okapi BM25

The Okapi BM25 (Robertson et al. 1994) ranking function uses term frequencies and inverse document frequencies as follows:

$$\text{Score}(Q,d) = \sum_{q \in Q} \log \frac{N - df(q) + 0.5}{df(q) + 0.5} \frac{tf(q,d)}{k_1((1-b) + b\frac{dl}{avdl}) + tf(q,d)} \tag{6}$$

where $N$ is the total number of documents in the collection, $df(q)$ is the number of documents containing the term $q$, $dl$ and $avdl$ are respectively the length of document $d$ and the average length of documents in the collection, and $b$, $k_1$ are parameters.

#### 3.1.5 doc2vec (*DV*)

Doc2vec (DV) is a document embedding method proposed by Mikolov et al. (2013a).

#### 3.1.6 Word Mover's Distance (*WMD*)

Kusner et al. (2015) proposed the Word Mover's Distance (WMD) to measure the dissimilarity between two documents $d$ and $d'$. To compute WMD, first each word in $d$, $d'$ is represented by its embedding. Then, each word in $d$ is "moved" to a word in $d'$ in total, or in parts. The cost of moving a word to another is measured by the Euclidean distance between their embeddings. Roughly, the minimum overall cost of moving all words in $d$ to words in $d'$ gives the WMD between $d$ and $d'$.

## 4 Experiments and Results

### 4.1 Datasets

For our experiments, we used the data provided by the First International Workshop on Exploitation of Social Media for Emergency Relief and Preparedness (SMERP)[3] held at ECIR 2017. The SMERP collection consists of 372,220 tweets / microblogs posted during the August 2016 earthquake in central Italy, collected over three days.

[3] http://www.computing.dcu.ie/~dganguly/smerp2017/

**Table 1** Performance of various methods

| Run Name | LM-JM $\lambda = 0.1$ | LM-D $\mu = 1.0$ | BM25 k=1.2, b=1.0 | WV | WV +MB | MB | WV SG | DV | WMD |
|---|---|---|---|---|---|---|---|---|---|
| MAP | 0.0519 | 0.0501 | 0.0447 | **0.0672** | 0.0397 | 0.0388 | 0.0260 | 0.0058 | 0.0010 |
| GMAP | 0.0499 | 0.0485 | 0.0428 | **0.0660** | 0.0396 | 0.0165 | 0.0180 | 0.0054 | 0.0002 |
| Rprec | 0.0929 | 0.0937 | 0.0946 | **0.1480** | 0.1131 | 0.1050 | 0.0958 | 0.0379 | 0.0133 |
| P@5 | 0.3000 | 0.1500 | 0.2500 | 0.2500 | 0.1000 | 0.1000 | 0.2000 | 0.1500 | 0.0500 |
| P@10 | 0.2750 | 0.2250 | 0.1750 | **0.2750** | 0.0500 | 0.1000 | 0.1500 | 0.1000 | 0.0500 |
| P@15 | 0.2500 | 0.1833 | 0.1667 | **0.2333** | 0.0833 | 0.1167 | 0.1500 | 0.1167 | 0.0333 |
| P@20 | 0.2000 | 0.1375 | 0.1375 | **0.2750** | 0.0875 | 0.1250 | 0.1625 | 0.1125 | 0.0500 |
| P@30 | 0.2000 | 0.1417 | 0.1000 | **0.2500** | 0.1167 | 0.1250 | 0.1500 | 0.0833 | 0.0500 |
| P@100 | 0.1575 | 0.1300 | 0.0850 | **0.2125** | 0.1375 | 0.1175 | 0.1125 | 0.0850 | 0.0250 |
| P@200 | 0.1250 | 0.1125 | 0.1000 | **0.1763** | 0.1225 | 0.1225 | 0.1237 | 0.0675 | 0.0288 |
| P@500 | 0.1570 | 0.1320 | 0.1270 | **0.1570** | 0.1040 | 0.1075 | 0.0945 | 0.0610 | 0.0220 |
| P@1000 | 0.1225 | 0.1240 | 0.1187 | 0.1390 | 0.0900 | 0.0968 | 0.0877 | 0.0565 | 0.0210 |

The best values are shown in bold font

The dataset also contains four queries and their relevance assessments. Each SMERP query has 4 fields:

- the query number tagged with <num> and </num>;
- a title (tagged using <title> and </title>), which is a very brief representation of the information need;
- a description (tagged with <desc> and </desc>), which is a somewhat more detailed specification of the information need; and
- a narrative (tagged with <narr> and <narr>), a thorough specification the information need, that spells out what is and is not relevant.

All four SMERP queries are provided in Appendix.

## 4.2 Experimental Setup

### 4.2.1 Pre-processing

Tweets and queries were pre-processed before indexing. The pre-processing steps are as follows:

1. URLs were removed.
2. Any token containing only punctuation was removed.
3. Any token containing only digits was removed.

**Table 2** Relevant tweets retrieved by semantic matching

| # Query | Tweet ID | Tweet text | Terms Exclusive to Tweet | Terms common between query and tweet |
|---|---|---|---|---|
| SMERP -T1 | 768418420 060745728 | Our disaster desk is active for earthquake activity today. Volunteers are monitoring. Volunteers please report in to Skype #hmrd via ĉt | disaster<br>earthquake<br>monitoring<br>skype | Volunteers |
| SMERP - T2 | 76855457 3787193344 | A sign of modern times - People urged to remove password from WiFi to help Italian earthquake relief https://t.co/8ObutXTmYb via mashable | Italian<br>earthquake | wifi |
| SMERP - T3 | 768397673 191931904 | I want to inform all my friends and people who have contacted me that in my area(VITERBO) there are not injured layers during the earthquake | friends<br>contacted<br>earthquake | inform<br>injured |
| SMERP -T4 | 76830642090 7466752 | All the @crocerossa emergency centers are mobilized at national level. Local volunteers are supporting affected people #earthquake | emergency<br>earthquake | Volunteers |

**Table 3** Language Model Jelinek-Mercer Smoothing (LM-JM) parameter $\lambda$ tuning using only SMERP data

| $\lambda =$ | MAP | GMAP | Rprec |
|---|---|---|---|
| 0.0 | 0.0157 | 0.0155 | 0.0538 |
| **0.1** | **0.0519** | **0.0499** | **0.0929** |
| 0.3 | 0.0507 | 0.0489 | 0.0929 |
| 0.5 | 0.0499 | 0.0482 | 0.0939 |
| 0.7 | 0.0494 | 0.0479 | 0.0958 |
| 0.9 | 0.0465 | 0.0453 | 0.0932 |

4. Stopwords were removed.
5. Finally, the rest of the tokens were stemmed using Porter's stemmer (Porter 1997).

After pre-processing, the SMERP collection contains 3,03,867 distinct words / tokens (Tables 1 and 2).

### 4.2.2 Baseline Methods

Parameters for the baseline methods (LM-JM, LM-D and BM25) were tuned on the test dataset itself. The reported performance of these baselines may thus be regarded as somewhat unrealistically high. Details of parameter tuning for these methods are given in Tables 3, 4 and 5.

### 4.2.3 Generating Word and Document Embeddings

We used a variety of corpora to generate word embeddings. We use the following labels for different variants of our basic approach (Section 3) that use embeddings generated from different sources / models.

- *WV*: uses embeddings generated from the SMERP collection using the continuous-bag-of-words model.
- *WVSG*: uses embeddings generated from the SMERP collection using the skip-gram model.
- *WV* + *MB*: uses continuous-bag-of-words embeddings generated from a combined corpus consisting of both the SMERP collection and the TREC 2011 Microblog Track collection (Ounis et al. 2011).
- *MB*: uses continuous-bag-of-words embeddings generated from only the TREC 2011 Microblog Track collection.

**Table 4** Language Model Dirichlet Smoothing (LM-D) parameter $\mu$ tuning using only SMERP data

| $\mu =$ | MAP | GMAP | Rprec |
|---|---|---|---|
| 0.5 | 0.0500 | 0.0484 | 0.0945 |
| **1.0** | **0.0501** | **0.0485** | **0.0937** |
| 100.0 | 0.0486 | 0.0477 | 0.0927 |
| 500.0 | 0.0430 | 0.0423 | 0.0867 |

**Table 5** Okapi BM25 (BM25) parameter $k_1$ and $b$ tuning using only SMERP data

| $k_1 =$ | $b =$ | MAP | GMAP | Rprec |
|---|---|---|---|---|
| **1.2** | **1.0** | **0.0447** | **0.0428** | **0.0946** |
| 1.4 | 1.0 | 0.0439 | 0.0419 | 0.0934 |
| 1.6 | 1.0 | 0.0433 | 0.0414 | 0.0922 |
| 1.8 | 1.0 | 0.0427 | 0.0406 | 0.0921 |
| 2.0 | 1.0 | 0.0421 | 0.0400 | 0.0912 |
| 1.2 | 1.2 | 0.0394 | 0.0366 | 0.0926 |
| 1.2 | 1.5 | 0.0394 | 0.0366 | 0.0926 |

Table 1 compares the performance of different methods. From the table, it is clear that *WV* outperforms other traditional IR methods as well as *WV+MB*, *MB*, *WVSG*, *DV* and *WMD* on all the evaluation measures. *WV*, the best method, achieved a MAP value of 0.0672 which is about 29% better than the second-best method (LM-JM). In terms of GMAP, the performance of *WV* is almost 32% superior to the second-best method (LM-JM). For both R-Prec and P@k, *WV* produces the best result for almost all the reported values of $k$. Once again, these results are considerably better than most of the baselines.

Since *WV* outperforms *WVSG* by a big margin, we did not generate skip-gram embeddings from the TREC 2011 Microblog Track collection or the merged collection.

From Table 1, we also see that word embeddings do not help if they are generated using tweets not related to the disaster period (*MB*, *WV* + *MB*). To be precise, the use of embeddings generated from *MB* or even *WV* + *MB* degrades the overall performance.

The superiority of the proposed method (variant *WV*) may be explained as follows. *WV* retrieves some relevant tweets where, apart from the query terms, there are some useful terms semantically related to the information need in the queries. These tweets were retrieved within the top 50 ranks by *WV*, but the other methods were unable to retrieve them even in the top 1000 ranks. For example, query SMERP-T1 seeks information about available resources. Thus, information about services like "free wifi", "sms", "calling facility" etc. is relevant. A relevant tweet (tweet id 768418420060745728) was retrieved at rank 37 by *WV*, but this tweet was not retrieved by the other models. This tweet has some terms such as "disaster", "earthquake", and "monitoring" that are important and semantically related to the query, although they are not query keywords. *WV* thus seems to be able to find semantically relevant tweets which contain important terms that are absent from the queries. Some more examples of a similar kind are given in Table 2.

## 5 Conclusion

Our proposed model *WV* comprehensively outperforms traditional information retrieval methods such as LM-JM and BM25, as well as other document embedding approaches such as DV and WMD. However, the absolute values of all metrics, even for the best performing method (*WV*) are very low. In the near future, we intend to investigate the reasons for this poor performance. We also need to study why the skip-gram model (used in *WVSG*) performed worse than the continuous-bag-of-words model (used in *WV*) for generating word embeddings. Finally, the poor performance of *DV* and *WMD* also needs looking into.

Our experiments suggest that embeddings generated using tweets outside the disaster time period hurt rather than improve performance. It is possible that better word embeddings may be learnt if more tweets can be collected from during the disaster period. We hope to explore this hypothesis in future work.

## Appendix: Queries in the SMERP Collection

<top>
<num>SMERP-T1</num>
<title> WHAT RESOURCES ARE AVAILABLE</title>
<desc>Identify the messages which describe the availability of some resources.</desc>
<narr> A relevant message must mention the availability of some resource like food, drinking water, shelter, clothes, blankets, blood, human resources like volunteers, resources to build or support infrastructure, like tents, water filter, power supply, etc. Messages informing the availability of transport vehicles for assisting the resource distribution process would also be relevant. Also, messages indicating any services like free wifi, sms, calling facility etc. will also be relevant. In addition, any message or announcement about donation of money will also be relevant.
However, generalized statements without reference to any resource would not be relevant.</narr>
</top>

<top>
<num>SMERP-T2 </num>
<title> WHAT RESOURCES ARE REQUIRED</title>
<desc>Identify the messages which describe the requirement or need of some resources.</desc>
<narr>A relevant message must mention the requirement / need of some resource like food, water, shelter, clothes, blankets, human resources like volunteers, resources to build or support infrastructure like tents, water filter, power supply, blood and so on. A message informing the requirement of transport vehicles assisting resource distribution process would also be relevant. Also, messages requesting for any services like free wifi, sms, calling facility etc. will also be relevant. In addition, messages asking for donation of money will also be relevant.
However, generalized statements without reference to any particular resource would not be relevant.</narr>
</top>

<top>
<num>SMERP-T3</num>
<title> WHAT INFRASTRUCTURE DAMAGE, RESTORATION AND CASUALTIES ARE REPORTED</title>
<desc>Identify the messages which contain information related to infrastructure damage, restoration and casualties</desc>
<narr>A relevant message must mention the damage or restoration of some specific infrastructure resources, such as structures (e.g., dams, houses, mobile towers), communication facilities (e.g., roads, runways, railway), electricity, mobile or Internet connectivity, etc. Messages reporting injury or death of people will also be relevant.
Generalized statements without reference to infrastructure resources would not be relevant.</narr>
</top>

<top>
<num>SMERP-T4</num>
<title> WHAT ARE THE RESCUE ACTIVITIES OF VARIOUS NGOs / GOVERNMENT ORGANIZATIONS</title>
<desc> Identify the messages which describe on-ground rescue activities of different NGOs and Government organizations.</desc>
<narr> A relevant message must contain information about relief-related activities of different NGOs and Government organizations engaged in rescue and relief operation. Messages that contain information about the volunteers visiting different geographical locations would also be relevant. Messages indicating that organizations are accumulating money and other resources will also be relevant. However, messages that do not contain the name of any NGO / Government organization would not be relevant.</narr>
</top>

## References

Bandyopadhyay, A., Ghosh, K., Majumder, P., Mitra, M. (2012). Query expansion for microblog retrieval. *IJWS*, *1*(4), 368–380. https://doi.org/10.1504/IJWS.2012.052535.

Corso, G.M.D., Gulli, A., Romani, F. (2005). Ranking a stream of news. In: WWW.

Diaz, F., Mitra, B., Craswell, N. (2016). Query expansion with locally-trained word embeddings. arXiv:1605.07891.

Dong, A., Chang, Y., Zheng, Z., Mishne, G., Bai, J., Zhang, R., Buchner, K., Liao, C., Diaz, F. (2010). Towards recency ranking in web search. In: WSDM, pp. 11–20. ACM. https://doi.org/10.1145/1718487.1718490.

Efron, M. (2010). Hashtag retrieval in a microblogging environment. SIGIR pp. 787–788. http://portal.acm.org/citation.cfm?id=1835449.1835616.

Ghosh, S., & Ghosh, K. (2016). Overview of the FIRE 2016 microblog track: Information extraction from microblogs posted during disasters. In: Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016., pp. 56–61. http://ceur-ws.org/Vol-1737/T2-1.pdf.

Ghosh, S., Ghosh, K., Chakraborty, T., Ganguly, D., Jones, G.J.F., Moens, M. (eds.) (2017). Proceedings of the First International Workshop on Exploitation of Social Media for Emergency Relief and Preparedness co-located with European Conference on Information Retrieval, SMERP@ECIR 2017, Aberdeen, UK, April 9, 2017, CEUR Workshop Proceedings, vol. 1832. CEUR-WS.org. http://ceur-ws.org/Vol-1832.

Hiemstra, D. (2000). Using language models for information retrieval. Ph.D. thesis, University of Twente.

Imran, M., Castillo, C., Diaz, F., Vieweg, S. (2015). Processing social media messages in mass emergency: A survey. *ACM Computing Surveys*, *47*(4), 67:1–67:38.

Ganesh, J., Gupta, M., Varma, V. (2016). Doc2sent2vec: A novel two-phase approach for learning document representation. In: SIGIR.

Jelinek, F., & Mercer, R.L. (1980). Interpolated estimation of markov source parameters from sparse data. In: Proceedings of the Workshop on Pattern Recognition in Practice.

Kim, H.K., Kim, H., Cho, S. (2017). Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, *266*(Supplement C), 336–352. https://doi.org/https://doi.org/10.1016/j.neucom.2017.05.046. http://www.sciencedirect.com/science/article/pii/S0925231217308962.

Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q. (2015). From word embeddings to document distances. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pp. 957–966. JMLR.org. http://dl.acm.org/citation.cfm?id=3045118.3045221.

Lau, J.H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv:1607.05368.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, pp. II–1188–II–1196. JMLR.org. http://dl.acm.org/citation.cfm?id=3044805.3045025.

MacKay, D.J., & Peto, L.C.B. (1994). A hierarchical dirichlet language model. *Natural Language Engineering*, *1*, 1–19.

Massoudi, K., Tsagkias, E., de Rijke, M., Weerkamp, W. (2011). Incorporating query expansion and quality indicators in searching microblog posts. *ECIR*, *2011*, 362–367.

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013b). In Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.) *Distributed representations of words and phrases and their compositionality*, (pp. 3111–3119). New York: Curran Associates, Inc. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Mikolov, T., Yih, W., Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In: NAACL HLT 2013.

Ounis, I., Macdonald, C., Lin, J., Soboroff, I. (2011). Overview of the trec-2011 microblog track. In: Proceeddings of the 20th Text REtrieval Conference (TREC 2011), vol. 32.

Ponte, J., & Croft, W. (1998). A language modeling approach to information retrieval. In: Proc. ACM SIGIR.

Porter, M.F. (1997). *Readings in information retrieval. chap. An Algorithm for Suffix Stripping*, (pp. 313–316). San Francisco: Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=275537.275705.

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M. (1994). Okapi at TREC-3. In: Proceedings of the Third Text REtrieval Conference (TREC 1994). NIST.

Varga, I., et al. (2013). Aid is out there: Looking for help from tweets during a large scale disaster. In: Proc. ACL.

Xing, C., Wang, D., Zhang, X., Liu, C. (2014). Document classification with distributions of word vectors. In: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, pp. 1–5. https://doi.org/10.1109/APSIPA.2014.7041633.

**Ayan Bandyopadhyay** has obtained M. Sc. Degree in Computer sciences. Pursuing his PhD from Jadavpur University from 2015. He was organizing committee member in FIRE evaluation campaigns (http://fire.irsi.res.in).

**Debasis Ganguly** is a researcher with 4+ years of post PhD experience working as a research staff member in IBM research, Dublin. Generally speaking, his research activities span the topics on Information Retrieval (IR) and Natural Language Processing (NLP). More specifically, he is interested in applying semantic relationships between text units (e.g. by embedding the text units in a vector space over reals) for improving various IR and NLP tasks. Some of his existing research work with word vector embedding has been shown to improve search and relevance feedback performance.

**Mandar Mitra** obtained a Bachelor's degree in Computer Science from the Indian Institute of Technology, Kanpur, and an MS and a PhD from Cornell University. He has been on the faculty of Indian Statistical Institute since 1999. He helps to coordinate the FIRE evaluation campaigns (http://fire.irsi.res.in).

**Sanjoy Kumar Saha** obtained his Bachelor and Master in engineering degree in Electronics and Tele-Communication from Jadavpur University, India in 1990 and 1992 respectively. Obtained PhD from Bengal Engineering and Science University, Shibpur (now IIEST, Shibpur), India in 2006. Currently working as Professor in Computer Science and Engineering Department of Jadavpur University, India. Research area includes signal processing, pattern recognition and information retrieval.

**Gareth J.F. Jones** obtained a B.Eng in Electrical and Electronic Engineering from the University of Bristol in 1989 and a PhD examining the Application of Linguistic Models in Continuous Speech Recognition in 1994 from the same institution. From 1993-96 he was a member of the Speech, Vision and Robotics Group, Department of Engineering and Computer Laboratory, University of Cambridge, working as a Research Associate on the Video Mail Retrieval using Voice (VMR) project. From 1996-2003 he was a Lecturer in Media Computing in the Department of Computer Science at the University of Exeter. From 1997-98 he was a Toshiba Fellow and engineer at the Toshiba Corporation Research and Development Centre in Kawasaki, Japan. In 2002 he was a Visiting Scientist to the Informedia project at Carnegie Mellon University, U.S.A. and a JSPS Visiting Fellow at the National Institute of Informatics, Toyko, Japan. In 2014 he was a Visiting Researcher at the Computer Laboratory, University of Cambridge In 2003 he was appointed as a Senior Lecturer in the School of Computing at Dublin City University, and promoted to Professor (Associate Professor) in 2015.