# Word Embeddings for Information Extraction from Tweets

Surajit Dasgupta*
Jadavpur University, India
surajit.techie@gmail.com

Abhash Kumar*
Jadavpur University, India
abhashmaddi@gmail.com

Dipankar Das
Jadavpur University, India
ddas@cse.jdvu.ac.in

Sudip Kumar Naskar
Jadavpur University, India
sudip.naskar@cse.jdvu.ac.in

Sivaji Bandyopadhyay
Jadavpur University, India
sivaji.cse.ju@gmail.com

## ABSTRACT

This paper describes our approach on "Information Extraction from Microblogs Posted during Disasters" as an attempt in the shared task of the Microblog Track at Forum for Information Retrieval Evaluation (FIRE) 2016 [2]. Our method uses vector space word embeddings to extract information from microblogs (tweets) related to disaster scenarios, and can be replicated across various domains. The system, which shows encouraging performance, was evaluated on the Twitter dataset provided by the FIRE 2016 shared task.

## CCS Concepts

**•Computing methodologies → Natural language processing; Information extraction;**

## Keywords

word embedding; information retrieval; information extraction; social media

## 1. INTRODUCTION

Social media plays a very important role in the dissemination of real-time information such as disaster outbreaks. Efficient processing of information from social media websites such as Twitter can help us to pursue proper disaster mitigation strategies. Extracting relevant information from tweets proves to be a challenging task, owing to their short and noisy nature. Information extraction from social media text is a well researched problem [3], [1], [9], [4], [8], [7]. Approaches using bag-of-words model, n-grams based methods and machine learning have been extensively used to extract information from microblogs.

## 2. TASK DEFINITION

A set of tweets, $T = \{t_1, t_2, t_3 \dots t_n\}$ and a set of topics, $Q = \{q_1, q_2, q_3 \dots q_m\}$ are given. Each topic contains a title, a brief description and a detailed narrative on what type of tweets are considered relevant to the topic. The tweets given in the task were posted during the Nepal earthquake[1] in April 2015. Each topic contains a broad information need during a disaster, such as – availability or requirement of general or medical resources by the population in the disaster affected area, availability or requirement of resources in a geographical region, reports of relief being carried out by an organization and reports of damage to infrastructure. The main objective of this task is to extract all tweets, $t_i \in T$ that are relevant to each topic, $q_j \in Q$ with high precision and high recall, and rank them in their order of relevance.

## 3. DATA AND RESOURCES

This section describes the dataset and resources provided to the shared task participants. A text file containing 50,068 tweet identifiers that were posted during the Nepal earthquake in April 2015, was provided by the organizers. A Python script was provided that downloaded the tweets using the Twitter API[2] into a JSON encoded tweet file which was processed during the task. A text file of topic descriptions in TREC[3] format was provided, that contained information necessary for the extraction of relevant tweets. The topic file consisted of 7 topics: FMT1, FMT2, FMT3, FMT4, FMT5, FMT6 and FMT7. Each topic consisted of the following 4 sections:

- <num> : Topic number.
- <title> : Title of the topic.
- <desc> : Description of the topic.
- <narr> : A detailed narrative which describes what types of tweets would be considered relevant to the topic.

## 4. SYSTEM DESCRIPTION

### 4.1 Preprocessing

We parsed the JSON encoded tweets and retrieved the following attributes – tweet identifier, tweet, geolocation. From the tweets, we removed the Twitter handles starting with @, URLs and all punctuation marks except the instances of a single " . " (period) and a single " , " (comma), using regular expressions. We removed the ASCII characters from the tweets and converted the remaining tweet to lower case characters.

We also preprocessed the <narr> sections of the topic file. We removed the punctuation marks, stop words and converted the text to lower case characters. The preprocessed <narr> sections for each topic was used for building the word bags.

*Indicates equal contribution.

[1] http://en.wikipedia.org/wiki/April_2015_Nepal_earthquake

[2] https://dev.twitter.com/overview/api/tweets

[3] http://trec.nist.gov

## 4.2 Word Bags

To build the topic-specific word bags, the preprocessed <narr> section was manually checked to retain the relevant words for each topic. The topic words were expanded using the synonyms obtained from NLTK WordNet[4]. The past, past participle and present continuous forms of verbs were obtained using the NodeBox[5] library for Python. Vowels, except the initial character, were removed to create unnormalized version of the words which are generally used in Twitter owing to the 140 character limit. The resultant set of words were used to create the word bags for each topic.

## 4.3 Entity Detection

For the topics FMT5 and FMT6, location and organization information was required to be detected from the tweet. To extract the location information, we used the *geo-location* attribute from the tweets and the Stanford NER tagger[6] to extract location names from the tweet text. Similarly, we used the Stanford NER tagger to detect organizations in the tweet text.

We split the tweet file into 10 files containing 5,000 tweets each. The Stanford NER tagger was used in parallel on the 10 splitted files to identify the location and organization, if any. This reduced the computation time by 85%.

## 4.4 Word Vectors

We used the pre-trained 200 dimensional GloVe [6] word vectors on Twitter data[7] (2 billion tweets) to create the vectors of the preprocessed tweets and the word bags.

The tweet vectors were created by taking the normalized summation of the vectors of the words in the tweets, which were present in the vocabulary of the pre-trained GloVe model. In cases where the word was not a part of the model vocabulary, it was assigned to the null vector.

$$\overrightarrow{\boldsymbol{t_i}} = \frac{1}{N_v\,(t_i)} \sum_{j=1}^{N_v(t_i)} \overrightarrow{\boldsymbol{u_{ij}}}$$

$$\text{and,}\quad \overrightarrow{\boldsymbol{u_{ij}}} = \vec{\boldsymbol{0}}, \quad \text{if } u_{ij} \notin \textit{vocabulary}$$

where,

$$\begin{aligned} \overrightarrow{\boldsymbol{t_i}} &= \text{Tweet vector of } i^{th} \text{ tweet, } t_i. \\ N_v\,(t_i) &= \text{Number of words in } t_i \text{ present in } \textit{vocabulary.} \\ \overrightarrow{\boldsymbol{u_{ij}}} &= \text{Vector of } i^{th} \text{ word in } j^{th} \text{ tweet.} \end{aligned}$$

Similarly, the word bag vectors were created by taking the normalized summation of the vectors of the words in the word bags, which were present in the vocabulary of the pre-trained GloVe model. Out of vocabulary words were assigned to the null vector.

$$\overrightarrow{\boldsymbol{q_i}} = \frac{1}{N_v\,(q_i)} \sum_{j=1}^{N_v(q_i)} \overrightarrow{\boldsymbol{w_{ij}}}$$

$$\text{and,}\quad \overrightarrow{\boldsymbol{w_{ij}}} = \vec{\boldsymbol{0}}, \quad \text{if } w_{ij} \notin \textit{vocabulary}$$

where,

$$\begin{aligned} \overrightarrow{\boldsymbol{q_i}} &= \text{Topic vector of } i^{th} \text{ word bag, } q_i. \\ N_v\,(q_i) &= \text{Number of words in } q_i \text{ present in } \textit{vocabulary.} \\ \overrightarrow{\boldsymbol{w_{ij}}} &= \text{Vector of } i^{th} \text{ word in } j^{th} \text{ word bag.} \end{aligned}$$

The tweet vector, $\overrightarrow{\boldsymbol{t_i}}$ and the word bag vector, $\overrightarrow{\boldsymbol{q_i}}$ are used to calculate the similarity.

The Word2Vec [5] library for Gensim[8] was used to create the tweet vectors and the topic vectors using the pre-trained GloVe model. The GloVe vectors were converted to Word2Vec vectors using code from the GitHub repository, `manasRK/glove-gensim`[9].

## 4.5 Similarity Metric

We used cosine similarity measure to calculate the cosine similarity, $S$ between the tweet vector and the topic vector.

$$\begin{aligned} S &= \textit{cosine-sim}(\overrightarrow{\boldsymbol{t_i}}, \overrightarrow{\boldsymbol{q_j}}) \\ &= \frac{\overrightarrow{\boldsymbol{t_i}} \cdot \overrightarrow{\boldsymbol{q_j}}}{||\overrightarrow{\boldsymbol{t_i}}||\ ||\overrightarrow{\boldsymbol{q_j}}||} \end{aligned}$$

A high value of $S$ denotes higher similarity between the tweet vector, $\overrightarrow{\boldsymbol{t_i}}$ and the topic vector, $\overrightarrow{\boldsymbol{q_j}}$ and vice versa.

For topics such as FMT5 and FMT6, where entity information such as location (LOC) or organization (ORG) was required, the consolidated score, $S'$ was calculated as follows:

$$S' = \frac{S + I}{2}$$

$$\text{where,}\qquad I = \begin{cases} 1, & \text{if LOC or ORG is present.} \\ 0, & \text{otherwise.} \end{cases}$$

The consolidated value, $S'$ shifts the cosine similarity towards 1 if the location or organization information is present (high relevance) and towards 0, otherwise (low relevance).

# 5. RESULTS AND ERROR ANALYSIS

Table 1 represents the results obtained by our word embedding based approach. As seen in the table, Run 1 has achieved the best results among the other runs, owing to the fact that Run 1 used word bags which were made from its corresponding descriptions for each topic, whereas the the other runs split the word bags categorically and averaged the similarity between the tweet vector and the split topic vectors.

| Run ID | Precision | Recall | MAP | Overall MAP |
|---|---|---|---|---|
| | @ 20 | @ 1000 | @ 1000 | |
| JU_NLP_1 | **0.4357** | **0.3420** | **0.0869** | **0.1125** |
| JU_NLP_2 | 0.3714 | 0.3004 | 0.0647 | 0.0881 |
| JU_NLP_3 | 0.3714 | 0.3004 | 0.0647 | 0.0881 |

**Table 1: Results of automatic runs**

The secondary performance obtained in Run 2, 3 is a result of the averaging which approximated the actual cosine similarity value between the tweet and topic vectors. Runs 2 and 3, which are identical in nature, used *cosine distance* as their similarity metric.

[4] http://www.nltk.org/howto/wordnet.html

[5] https://www.nodebox.net/code/index.php/Linguistics

[6] http://nlp.stanford.edu/software/CRF-NER.shtml

[7] http://nlp.stanford.edu/projects/glove/

[8] https://radimrehurek.com/gensim/models/word2vec.html

[9] https://github.com/manasRK/glove-gensim

## 6. CONCLUSION

In this paper, we presented a brief overview of our system to address the information extraction from microblog data. We have observed that, building word bags which contained all the topic words relevant to the topic showed better results than splitting the word bags. Therefore, Run 1 exhibited better results than the rest. Considering *hashtags* as a feature should also improve the performance of the system.

As a future work, we work like to explore more sophisticated techniques to build the vectors of the tweets, given the vectors of its constituent words, by considering the sequence of the words into account. We also plan to incorporate more topic specific features to improve the performance of our system.

## 7. REFERENCES

[1] S. Choudhury, S. Banerjee, S. K. Naskar, P. Rosso, and S. Bandyopadhyay. Entity extraction from social media using machine learning approaches. In *Working Notes in Forum for Information Retrieval Evaluation (FIRE 2015)*, pages 106–109, Gandhinagar, India, 2015.

[2] S. Ghosh and K. Ghosh. 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*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[3] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier. Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, pages 1021–1024, New York, NY, USA, 2013. ACM.

[4] M. Imran, S. M. Elbassuoni, C. Castillo, F. Diaz, and P. Meier. Extracting information nuggets from disaster-related messages in social media. *Proc. of ISCRAM, Baden-Baden, Germany*, 2013.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[6] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[7] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 841–842, New York, NY, USA, 2010. ACM.

[8] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1079–1088. ACM, 2010.

[9] X. Yang, C. Macdonald, and I. Ounis. Using word embeddings in twitter election classification. *arXiv preprint arXiv:1606.07006*, 2016.