Exploiting Neural Embeddings for Social Media Data Analysis

Sadid A. Hasan, Yuan Ling, Joey Liu, and Oladimeji Farri
Artificial Intelligence Laboratory, Philips Research North America, Cambridge, MA, USA
{sadid.hasan,yuan.ling,joey.liu,Dimeji.Farri}@philips.com

Abstract

In this paper, we describe our microblog real-time filtering system developed and submitted for the Text Retrieval Conference (TREC 2015) microblog track. We submitted six runs for two tasks related to real-time filtering by using various Information Retrieval (IR), and Machine Learning (ML) techniques to analyze the Twitter sample live stream and match relevant tweets corresponding to specific user interest profiles. Evaluation results demonstrate the effectiveness of our approach as we achieved 3 of the top 7 best scores among automatic submissions across all participants and obtained the best (or close to best) scores in more than *25%* of the evaluated topics for the real-time mobile push notification task.

1 Introduction

The main goal of the real-time filtering task in the TREC 2015 microblog track[1] was to monitor a stream of social media posts (Twitter) in order to match relevant content (tweet) corresponding to specific user interest profiles and push notifications to the users considering two different tasks. For *Task A*, a maximum of 10 interesting tweets per day were requested to be sent to the user in real-time, whereas in *Task B*, a batch of up to 100 top ranked interesting tweets per interest profile were required to be sent at the end of each day. We submitted six runs for the two tasks using various Natural Language Processing (NLP), specifically IR techniques to analyze the Twitter sample live stream and match relevant tweets corresponding to specific user interest profiles.

[1]https://github.com/lintool/twitter-tools/wiki/TREC-2015-Track-Guidelines

NLP and ML have emerged in the recent decades as prominent methodologies to fully leverage the rapid explosion of unstructured information in the digital universe. Deep Learning (DL) techniques typically aim at automatically learning representations of data without requiring any prior domain knowledge or expert annotations. DL alleviates the need for tedious feature engineering by devising efficient unsupervised or semi-supervised algorithms for hierarchical feature learning and extraction. Deep learning for NLP tasks mainly rely on learning high-dimensional vector representations of words, phrases, sentences, or documents and their relationships (called *embeddings*) using neural network architectures. The trained language model transforms semantically similar textual units into similar vector representations (Mikolov et al., 2013; Le and Mikolov, 2014). The main advantage of such architecture over the traditional bag-of-words model is its ability to capture the embedded ordering and semantics by learning fixed-length vector representations for variable-length text structures. We exploited the strength of neural word and phrase embeddings in extending the context of the underlying user interest profiles for our microblog real-time filtering system. In the subsequent sections, we describe the overall architecture of our system, and present the evaluation results.

2 System Description

Figure 1 shows the generic architecture of our real-time content filtering system. Our overall approach comprises three main processes: (i) Analysis of Interest Profiles: leveraging a topic signature modeling algorithm and neural word/phrase embeddings

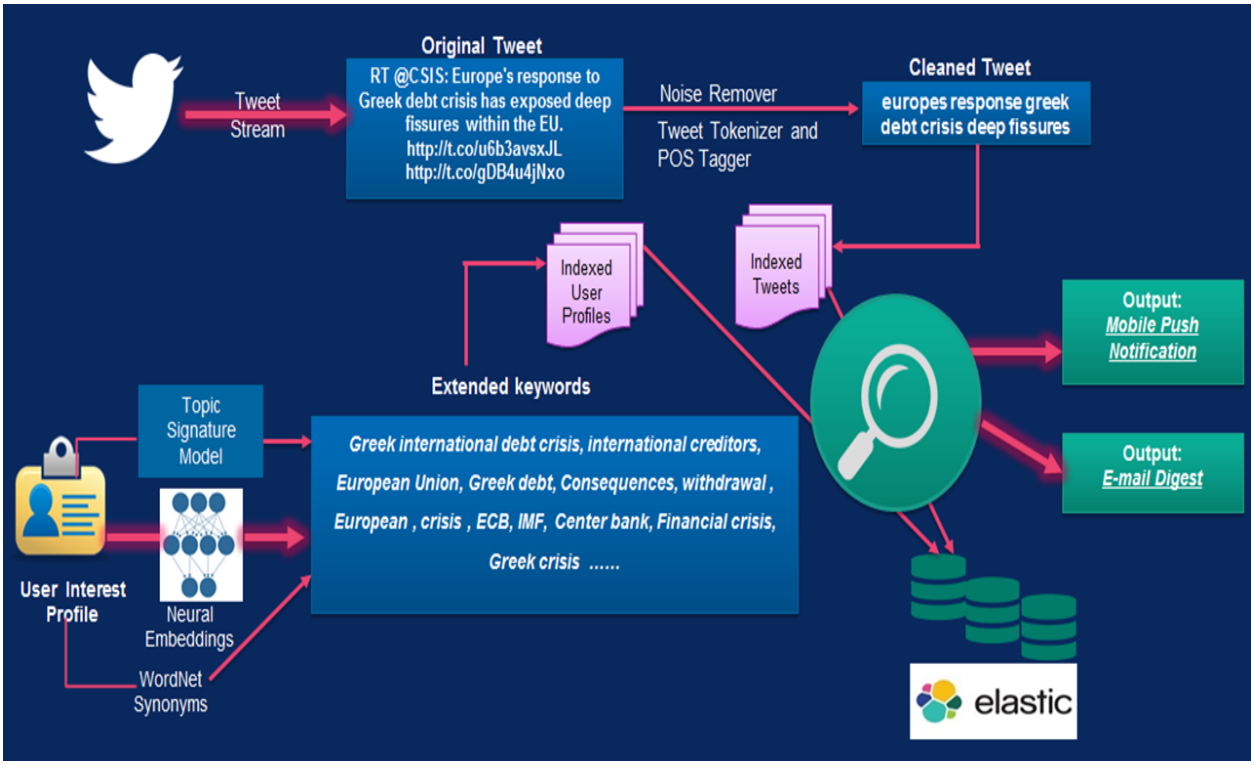

Figure 1: Real-time filtering system architecture

for contextual understanding, (ii) Tweet Content Analysis: noisy element filtering, tokenization and Parts-Of-Speech (POS) tagging for generation of a cleaned version of the tweet, and (iii) Relevant Content Matching: mapping of relevant tweets to corresponding interest profiles using a weighted combination of a term frequency-inverse document frequency (TF-IDF)-based content matching score and a semantic similarity score. The submitted runs are varied based on different context expansion methodologies used during topical analyses of interest profiles along with different threshold values for the tweet relevance score.

Initially, we analyzed all user interest profiles provided by the track organizers using a topic signature algorithm (Lin and Hovy, 2000) to extract the most important topical keywords to capture the overall context of the information need. These keywords along with their n-gram combinations were utilized to expand the topical vocabulary by extracting related synonym sets from the WordNet database (Fellbaum, 1998) and exploiting deep neural word/phrase embeddings. The neural word/phrase embeddings were trained on over 60 million tweets (crawled from the Twitter's live sample stream for over two weeks before the start of the evaluation period) by using a deep learning-based word/phrase to vector representation modeling algorithm (Mikolov et al., 2013). The expanded topical keyword list per interest profile was indexed in Elasticsearch[2] for further analyses during real-time tweet content filtering.

In the second step, each incoming tweet was processed to remove noise using various rule-based algorithms in association with curated databases of known noisy elements that are widely used in tweet messages. We then applied tokenization and part-of-speech (POS) tagging (Gimpel et al., 2011) to extract the most important words that preserve the contextual meaning of the tweet.

In the last step, the tweet content words were used as query words for which an appropriate interest profile was retrieved and matched using algorithms within Elasticsearch. Elasticsearch transformed the query words as various combinations of

[2]http://www.elasticsearch.org/

possible n-grams/phrases to find an overall content match across all the interest profiles. The final relevance of a tweet with respect to a user interest profile was measured using a weighted combination of two scores (Eq 1): (i) TF-IDF based content matching score returned by Elasticsearch, and (ii) semantic similarity score based on an algorithm built on semantic networks of related words and corpus-based statistics (Li et al., 2006).

$$W * semantic_{score} + (1 - W) * elastic_{score} \quad (1)$$

We also calculated the semantic similarity of a new tweet with the tweets that were already sent to the users to minimize redundancy. Finally, an average relevance score over a set of empirical threshold values triggered a tweet to be sent to the matching user for *Task A* (within a few seconds after the tweet was originally created). For *Task B*, the incoming tweets were indexed through the end of each day, processed using the same algorithms as *Task A*, and then the interest profile keywords were used as queries to find a ranked list of up to 100 matching tweets to be sent as an e-mail digest to the corresponding user.

3 Experimental Setup

3.1 Test Data

The test dataset comprises 225 user interest profiles with three fields: "title" contained a few keywords, the "description" contained a one-sentence statement of the information need, and the "narrative" was a paragraph-length description of the information need. We used all fields for our experiments.

3.2 Corpus

Twitter's live tweet sample stream was used as the corpus for the track. We built an architecture to continuously monitor the tweet stream by following the guidelines provided by the organizers.

3.3 Run Description

For *Task A*, we submitted three runs as follows: 1) prna1-A: considers topical keyword expansion using WordNet synonyms, 2) prna2-A: considers topical keyword expansion using neural word/phrase embeddings, and 3) prna3-A: considers topical keyword expansion using both WordNet synonyms and neural word/phrase embeddings. The runs were also varied by the threshold values for the tweet relevance scores as we set the highest threshold for run 3 and the lowest for run 1. Our three runs for *Task B* were designed in the same fashion except we use the Elasticsearch score as the only measure for tweet relevance due to time complexity and chose up to 100 top ranked tweets per profile to be sent to the user after the end of each day.

3.4 Evaluation and Analysis

The evaluation of the 2015 microblog track was conducted on a subset of 51 topics (selected out of the total 225 test topics) by following a similar procedure as the tweet timeline generation (TTG) task from the TREC 2014 microblog track (Wang et al., 2015; Lin et al., 2014). The tweets returned by the participant systems were collected into a single judgment pool for both tasks and each tweet was judged by the assessors independently corresponding to the user interest profiles using a three-point scale where spam/junk/not interesting, somewhat interesting, and very interesting tweets received the gain (score) of 0, 0.5, and 1, respectively. After this standard pooling assessment procedure, a clustering protocol was applied to group all tweets into a set of semantically similar clusters. The participant systems were only credited for returning one tweet from each meaningful cluster.

Figure 2 and Figure 3 show the overall scores of our runs for *Task A* across the selected 51 topics as compared to the *max* scores among all participants' submitted runs for two evaluation measures: expected latency-discounted gain (ELG[3]), and normalized cumulative gain (nCG[4]). These results show that our system achieves the best scores across all runs for 14% of the evaluated topics while having close to best scores for 12% of the evaluated topics. In-depth analyses of the results also reveal that WordNet synonyms and neural word/phrase embeddings often have a positive impact on the tweet relevance scores. Figure 4 shows the distribution of per-topic ELG scores for the best runs by mean ELG scores (adopted from the TREC 2015 overview

[3] *ELG* is computed using the summation of tweet gains normalized by the number of tweets returned.

[4] *nCG* is calculated by the total tweet gain divided by the maximum possible gain given the 10 tweet per day limit.

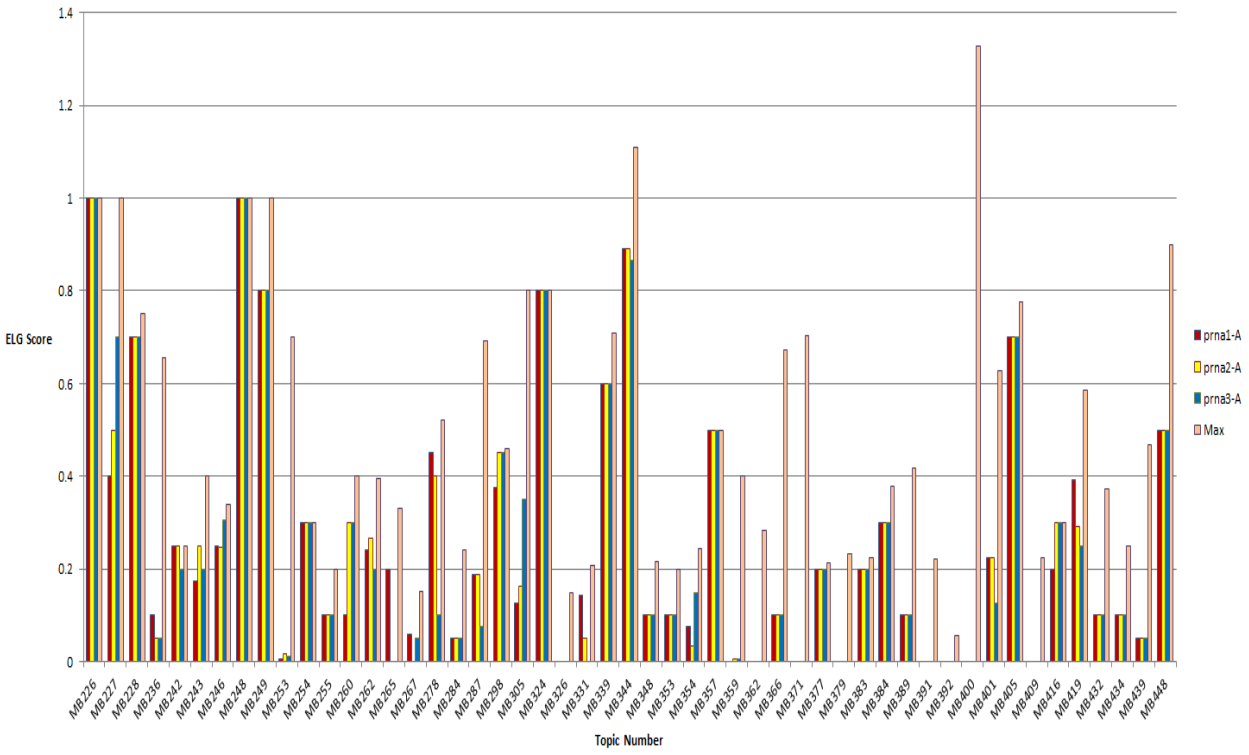

Figure 2: ELG scores for each topic (Task A)

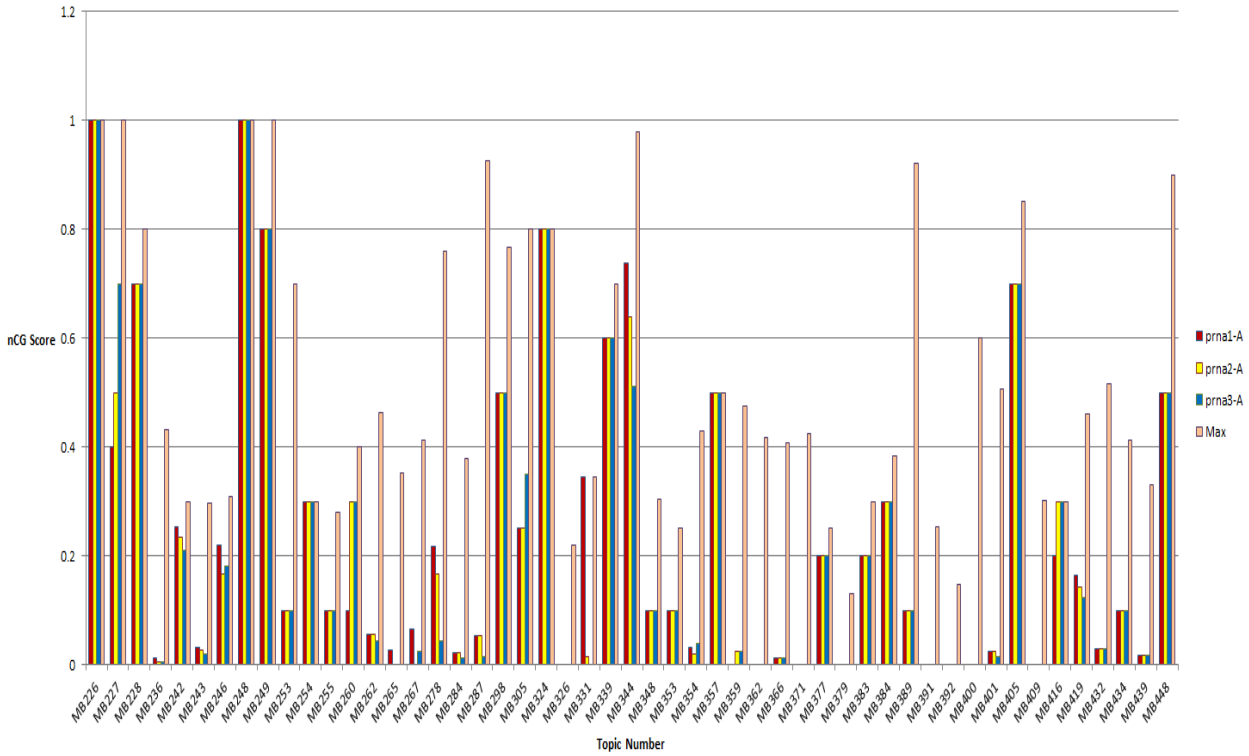

Figure 3: nCG scores for each topic (Task A)

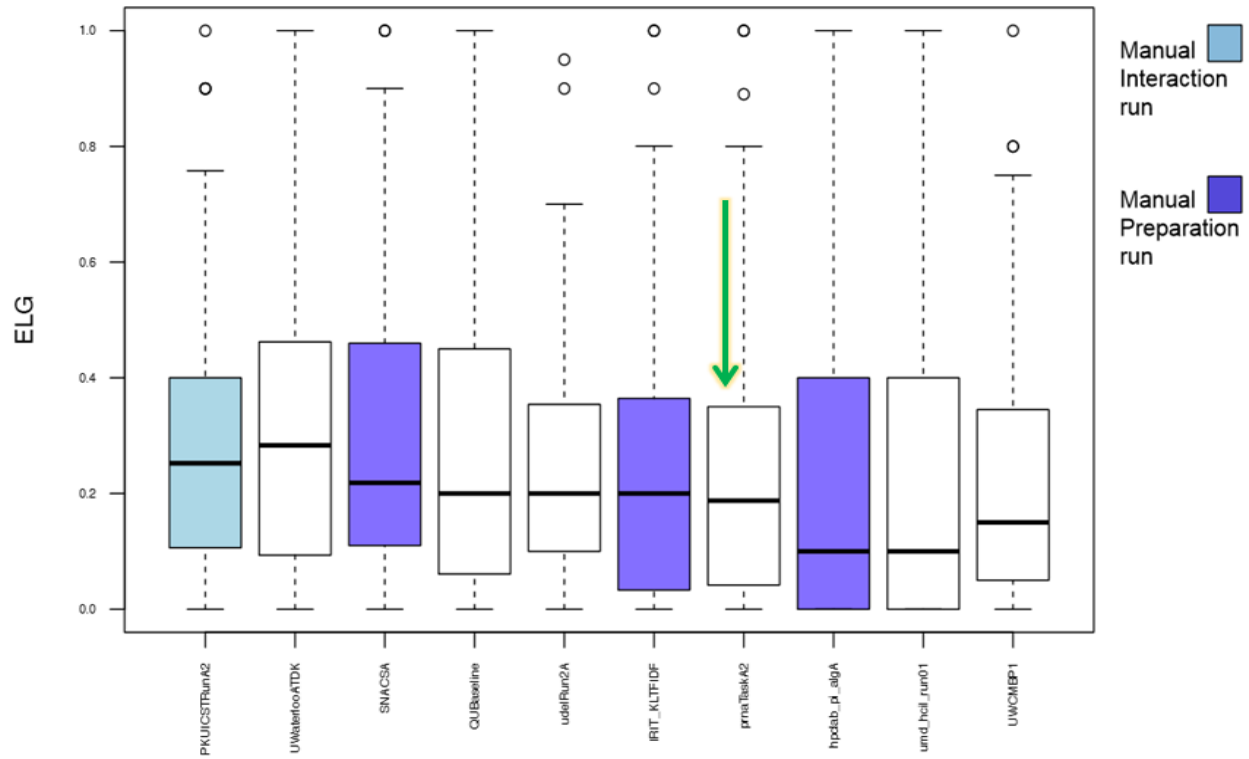

Figure 4: Distribution of per-topic ELG scores for best run by mean ELG (Task A); the arrow denotes our system

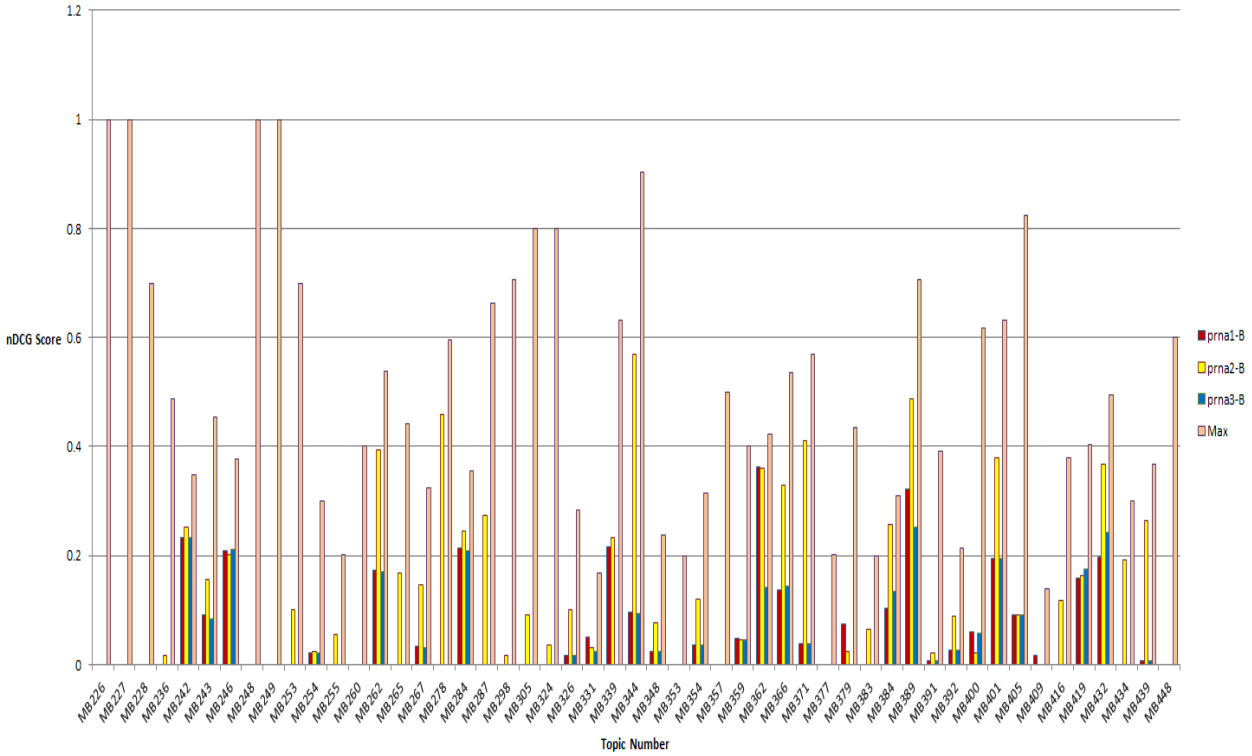

Figure 5: nDCG scores for each topic (Task B)

talk). Analyses reveal that our run (prnaTaskA2, arrow marked in Figure 4), which exploits neural embeddings for better understanding of the user interest profiles, has achieved one of the top 5 scores among all automatic runs submitted by the participants. Further analyses denote that all three of our submitted runs for *Task A* are placed in the best 7 ranks across all automatic submissions.

Figure 5 shows the overall scores of our runs for *Task B* across the selected 51 topics as compared to the *max* scores among all participants' submitted runs for the evaluation measure: normalized Discounted Cumulative Gain (nDCG), which computes the quality of ranking for each system based on the ranked list of 100 tweets graded with relevance judgment and normalized by the maximum possible gain. These results demonstrate that our system can achieve close to the best scores for a few number of topics simply because we could not implement the semantic similarity measure to compute the tweet relevance due to time complexity limitation. Contextual expansion methodologies (i.e. use of WordNet synonyms, and neural embeddings) show a similar positive effect on the overall results as shown in *Task A*.

4 Conclusion and Future Work

In this paper, we described our participation in the TREC 2015 microblog track. Evaluation results showed the effectiveness of our approach as we achieved additional gains with the implementation of neural word and phrase embeddings in extending relevant contexts for the user interest profiles. In future, we plan to improve upon our tweet relevance scoring algorithms, especially for the email digest task (B), by leveraging powerful computational resources to solve the respective use cases.

References

C. Fellbaum. 1998. WordNet - An Electronic Lexical Database. Cambridge, MA. MIT Press.

K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 42–47. ACL.

Q. V. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150.

C. Lin and E. H. Hovy. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. In *Proceedings of the 18th conference on Computational linguistics*, pages 495–501.

J. Lin, M. Efron, Y. Wang, and G. Sherman. 2014. Overview of the TREC-2014 microblog track. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC 2014, Gaithersburg, Maryland, USA, November 19-21, 2014*.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems NIPS 2013*, pages 3111–3119.

Y. Wang, G. Sherman, J. Lin, and M. Efron. 2015. Assessor differences and user preferences in tweet timeline generation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 615–624.