Iterative Keyword Optimization

Aviad Elyashar, Maor Reuben, and Rami Puzis

Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev,
Telekom Innovation Laboratories,
{aviade,maorreu}@post.bgu.ac.il, puzis@bgu.ac.il,

Abstract. Short keyword queries are one of the main tool of any user or bot seeking information through the ubiquitous search engines. Automated keyword optimization relies primarily on the analysis of data repositories in order to find a small set of keywords that identify the topic discussed and relevant documents. However, most search engines, available today on the Web are opaque, providing little to no information about their methods and the searched repository.
In this paper, we propose an automated iterative optimization of short keyword queries in order to improve information retrieval from opaque (black box) search engines. The use case considered involves the retrieval of relevant posts from online social media for a given a news article (claim) discussed online. The proposed algorithm iteratively selects keywords while querying the search engine and comparing a small set of retrieved posts to the given news article using a *mean relevance error* based on word embedding. We demonstrate the proposed algorithm while building a *Fake News* dataset from claims (collected from fact-checking websites) and their associated tweets. The proposed *mean relevance error* was found to be accurate for differentiating between *relevant* and *irrelevant* posts (0.9 AUC). The optimized queries produce similar results to manually extracted keywords and outperform TF-IDF based methods and POS tagging.

Keywords: Information Retrieval. · Opaque Search Engine. · Fake News.

1 Introduction

Short keyword queries are one of the main tool of any user or bot seeking information through the ubiquitous search engines available on the Web [3]. In this paper, we focus on retrieving posts from online social media (OSM). Search within OSM can help with box office revenues prediction [13], product reviews [6], and many other problems, where the intelligence of the crowd can be utilized. Unfortunately, in many cases, the ambiguity of short keyword queries causes poor performance [4]. The problem of ambiguity is more emphasized when working with opaque search engines such as Twitter search engine or Google search engine. In contrast to transparent search engines where the repository and algorithms are visible to the query writer, opaque search engines provide a very limited level of interactivity and hide all activities performed by the search engine, as well as the repository itself [7]. Today, most search engines and OSM are opaque.

In recent years, one of the OSM search use cases involves, the issue of fake news. In recent years, we have witnessed an enormous increase in the amount of fake news, disinformation, and propaganda on OSM, leading to the erosion of public trust in media outlets and OSM [25]. Several researchers have suggested investigating and determining the trustworthiness of a claim made in public media based on data extracted from OSM [19,18]. Such investigations require collecting posts associated with (presumably fake) claims that appear in news articles. In these approaches, the set of keywords used to query on OSM are defined manually for each and every claim. Assigning keywords manually significantly limits the number of claims that can be processed using the techniques proposed above.

In this paper, we suggest an automated iterative keyword optimization method for improving information retrieval from opaque search engines. The specific task being considered in this paper is the retrieval of relevant posts from OSM for a given news article or document being discussed online (referred to as a *claim*). We propose the following two-step solution: (1) The relevance of posts to the claim is estimated by comparing their vector representations; (2) A greedy algorithm that iteratively queries the OSM while optimizing the result relevance estimation is used. We demonstrate the proposed approach on Twitter while building a *Fake News* dataset[1] of 398 claims collected from fact-checking websites, as well as the word embedding of 1,186,334 posts relevant to those claims.

[1] https://drive.google.com/drive/folders/1yMfHexodEHa03D2BBF6bD6aeSXMREHJe

The rest of the paper is organized as follows: In Section 2, we review previous approaches for document similarity, keyword expansion, and active retrieval methods. In Section 3, we present the proposed solution which is composed of two complementary methods. The suggested *mean relevance error* is described in Section 3.1, and the greedy search algorithm is described in Section 3.2. In Section 4, we present the dataset we used for evaluating the solution, including the process of manually labeling tweets (see Section 4.1). In Sections 4.2 and 4.3, we present the results of the evaluation process on both methods. Section 5 discusses ethical considerations, and we conclude the paper in Section 6 with our plans for future work.

2 Related Work

In recent years, various solutions have been suggested for estimating the relevance of results provided by search engines and how to retrieve relevant results. In this section, we review studies related to the issues we focused on in this study.

2.1 Document Similarity

Studies in this domain evaluate the semantic similarity between two given documents. Several methods used word vector representation for this problem. In [9] calculated the minimal distance between each word in one text and all of the words in an other. [8] also used these vectors for calculating the distances between words in documents. Based on these distances they determined whether the documents were similar or not. In this study we use the latter method [9] to estimate the the relevance of results. However, in contrast to that, we evaluate the similarity between a given document (claim) to a collection of posts (documents) associated with the given claim.

2.2 Keyword Expansion

In the last few decades, the ambiguity of short keyword queries resulted in the need for improved solutions for the Web retrieval task [3]. A common approach for keyword expansion takes given keywords and adds other related words for better representation. In prior research synonyms from WordNet[2] were added to improve keyword representation of text [21,20]. In [1], the authors showed that Wikipedia can be a source for keywords expansion. A novel part of speech (POS) patterns that can be used for choosing candidate keywords were presented in [12]. Similarly, in [21] the authors used the TF-IDF measure for keywords expansion, by simply choosing the K best terms based on the TF-IDF score. Kuzi et al. [10] proposed a method for choosing a term for query expansion using word embedding representation of terms. In this case, the idea is to choose terms that yield the highest probability of being related to the current query.

Pseudo relevance feedback, a retrieval technique that extracts words from the results of a given query, is another method of query expansion. This approach is applied in order to reduce the semantic gap between the query and the results. Zamani et al. [23] refer to query expansion as a recommendation task, and used matrix factorization for recommending expansion terms. Xu et al. [22] presented a framework that selects expansion terms using a modified term-dependency method. The framework generates term features for the candidate terms based on word representations to encode relationships between given queries and corresponding candidate terms.

2.3 Active Retrieval

In the active retrieval domain, the system actively selects documents for user feedback in order to improve the precision or recall of results. For instance, ReQ-ReC [11] is a double-loop active retrieval system that combines iterative expansion of a query set to improve recall with iterative refinements of a classifier to improve precision. ATR-Vis, a user-driven visual approach for the retrieval of Twitter content [14] is another active retrieval system. ATR-Vis uses four strategies of active learning in order to improve retrieval of relevant tweets. The drawback of active retrieval systems is based on their need for user feedback in order to improve the relevance of the retrieved posts.

[2] https://wordnet.princeton.edu/

3 Methods

In this paper, we propose a novel iterative approach for optimizing keyword extraction given a document while querying a search engine. The proposed approach includes two complementing methods: First, in Section 3.1, we discuss the *mean relevance error* – a short-document comparison method for determining the relevance of query results to a given document based on estimating the minimal distance between words comprising both the retrieved posts and the input document. Second, in Section 3.2, we suggest different iterative searches for finding the most appropriate keywords in order to retrieve the maximal number of relevant posts using an opaque search engine. Since we have no knowledge about the inner mechanisms of the search engine and the data stored there, we perform series of limited interactions with the search engine in order to optimize the set of keywords comprising the query.

3.1 Mean Relevance Error

In this section, we discuss the *mean relevance error* – a document comparison method for determining the relevance of query results to a given document based on estimating the minimal distance between the document words and the results words.

First, we remove stop words from the document and retrieved posts. The *mean relevance error* (MRE) is defined as a function which receives a document d and collection of posts P retrieved from the search engine as input and outputs a number. The lower the MRE is the more relevant the retrieved posts P are to the document d. $W_p = \{w_1, w_2, \ldots, w_k\}$ is the set of words in $p \in P$, and $W_d = \{w_1, w_2, \ldots, w_l\}$ denotes the set of words in the input document d. Since we focus on the retrieval of microblog posts which are relevant to some online discussions, we assume that $l \gg k$. As a measure of similarity between two words, we use the Euclidean distance between vector representations of the words, denoted by $dist(w_i, w_j)$. Vector representations of words can be derived using any word embedding model, such as GloVe [17], Word2vec [15], fastText [2], etc. The distance between a word w_i and a document d is the minimal distance between the word w_i and all of the words in W_d.

$$dist(w_i, d) = \min_{w_j \in W_d} \{dist(w_i, w_j)\}$$

Next, in order to calculate the distance of a post p from a document d, we average the distances of all words $w_i \in W_p$ to the document d:

$$dist(p, d) = \frac{1}{k} \sum_{w_i \in W_p} dist(w_i, W_d)$$

Finally, given a collection of posts P, we define the *mean relevance error* (MRE) of the collection of posts P to the document d as the average distance of all posts in P from d:

$$MRE(P, d) = \frac{1}{|P|} \sum_{p \in P} dist(p, d)$$

The MRE defined above is designed to measure only the relevance of the results. Other important aspects, such as the number of results, are intentionally not captured by the MRE. The quality of the MRE is affected by the quality of the underlying word embedding model. For general purpose query evaluation, using word embedding models trained globally on large datasets is recommended.

3.2 Keyword Optimization

In this section, we propose a novel automated approach for finding the most appropriate keywords in order to retrieve the maximal number of relevant documents when using an opaque search engine. The proposed method is based on an iterative greedy search for the best word to add to the input query in order to improve the relevancy posts retrieved by the search engine.

Algorithm 1 Bottom-Up Search

```
procedure BOTTOM-UP(DOCUMENT, MINPOSTS)
    $WalkedQueryList \leftarrow []$
    $baseQuery \leftarrow$ ""
    $keywords \leftarrow set(splitToKeywords(document))$
    $se = SearchEngine$
    while $keywords.size > 0$ do
        $query \leftarrow baseQuery$
        $bestWord \leftarrow$ ""
        $bestRelevance \leftarrow \infty$
        for all $keyword \in keywords$ do
            $query.add(keyword)$
            $posts \leftarrow se.getPosts(query)$
            $MRE \leftarrow calculateRelevance(document, posts)$
            if $posts.size() > minPosts$ then
                $WalkedQueryList.add(query, MRE)$
            if $MRE < bestRelevance$ then
                $bestRelevance \leftarrow MRE$
                $bestWord \leftarrow keyword$
        if $bestWord ==$ "" then
            $bestWord \leftarrow keywords[0]$
        $baseQuery.add(bestWord)$
        $keywords.remove(bestWord)$
    $bestQuery \leftarrow MinMRE(WalkedQueryList)$
     return $bestQuery$
```

Bottom-Up First, we split the given document's text into a set of words and remove stop words. In the first iteration, we start from queries with a single word. Each query is sent to the opaque search engine and receives posts as a response. Each keyword receives an aggregated *mean relevance error* (MRE), which reflects the relevance of the retrieved collection of posts to the given document. At the end of the iteration, we add the keyword with the highest MRE. The iterative process ends when the query includes all of the document's keywords or when the MRE doesn't decrease between iterations. The algorithm returns the query that yields the best MRE (see Algorithm 1). In this paper, we suggest and evaluate the proposed MRE (see Section 3.1). However, every other score can be suggested as an optional *relevance measure*.

4 Evaluation

This section includes a description of the data collection process using the methods presented in Section 3, as well as the evaluation results.

4.1 Data

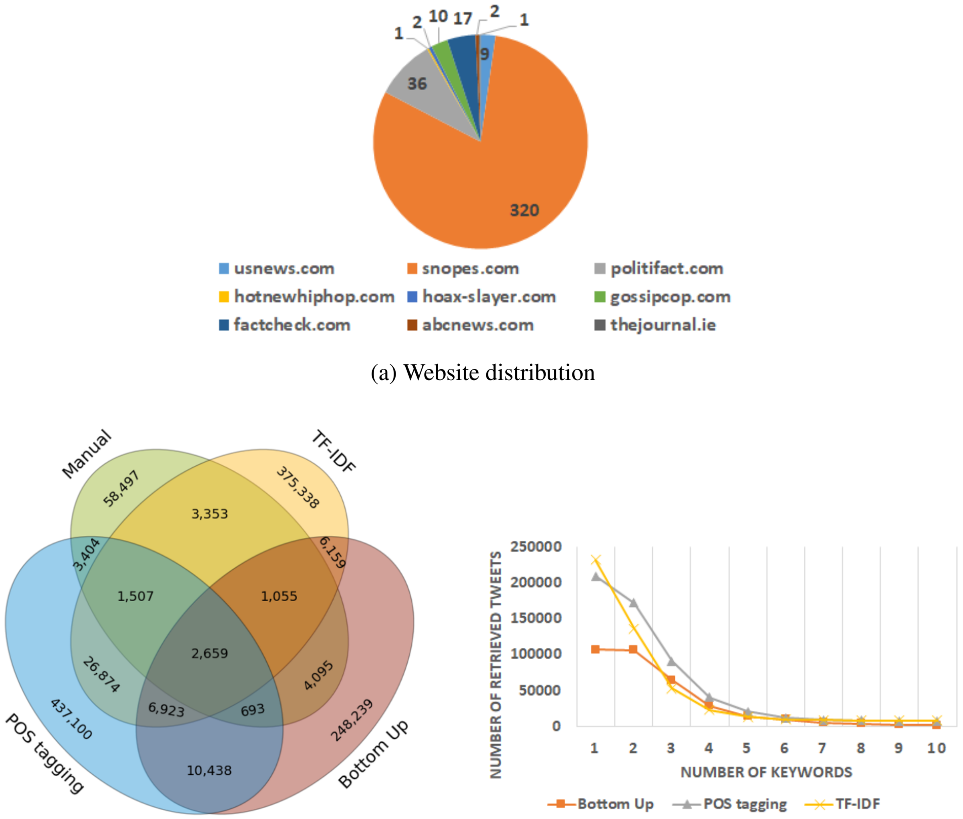
We collected 398 labeled claims from several fact-checking websites, primarily Snopes[3], as seen in Figure 1a. The claims were published from June 1997 to December 2018. Each claim includes descriptive attributes, such as title, description, verdict date (the date in which a fact-checker published the claim), etc. We used the Twitter search engine in order to collect tweets relevant to these claims. In total, we retrieved 1,186,334 tweets published by 772,940 users (with an average of 2,981 posts per claim). All of the tweets were published from April 2007 until February 2019. These tweets were crawled using four different methods: the proposed *Bottom-Up* greedy search (280,261 tweets), keywords defined manually (75,263 tweets), TF-IDF (423,868 tweets), and part of speech (POS) tagging (489,598 tweets), as can be seen in Figure 1b. For the keywords defined manually and the TF-IDF and POS tagging methods, we collected tweets by querying a different number of unique words (from one to ten), as seen in Figure 1c.

Manual Labeling of Tweet Relevance After retrieving tweets according to keyword suggestions, we labeled a subset of tweets for evaluation. In order to construct an evaluation dataset for each claim we set the maximal number of retrieved tweets at 40 tweets using the

[3] https://www.snopes.com/

keywords obtained manually and TF-IDF keywords. We calculated the MRE measure for all of the claims. Later, we sorted the claims by the MRE measure and collected the five claims with the maximal and minimal MRE (in total ten claims). We chose 20 claims that obtained the maximal and minimal MRE. The retrieved tweets also included retweets, and tweets that include links to news websites. The same process was conducted using tweets that are retrieved based on manual keyword assignment and using tweets that are based on TF-IDF keywords.

For the 20 claims, we collected a total of 1,173 related tweets. In the labeling process, we used three annotators (students) who were required to read the claim's title and description and the retrieved tweets associated with it. Then each annotator labeled each tweet with one of the optional labels: *Relevant* for cases in which the given tweet is associated with the given claim, *Irrelevant* for cases in which the give tweet is not associated with the given claim, and *Unknown* for cases in which the annotator is not certain whether the tweet is related or not. We only used the tweets that the majority of annotators agreed on (1,078 tweets of the original 1,173 retrieved tweets). For an example of a claim and *relevant*, and *irrelevant* tweets see Table 1.



(a) Website distribution

(b) Number of retrieved tweets for each keyword extraction method

(c) Number of retrieved tweets per number of keywords in a given query

Fig. 1: Data statistics

Table 1: Example of the labeled tweets associated with a given online discussion

Claim	"The rapper DMX (Earl Simmons) died in February 2018." (Fake)
Relevant Tweet	"Juan is just think DMX died so good time!"
Irrelevant Tweet	"I liked a @ YouTube… video DMX - I Just Died in your arms Tonight. [Remix]"

4.2 Mean Relevance Error

In order to evaluate the proposed method, we defined the following experimental setup. For word embedding, the word vectors representations were obtained by a pre-trained word embedding model of fastText [16]. The model was trained on Common Crawl [4] and Wikipedia [5] using the fastText library [6]. We used Euclidean as the distance measure.

We evaluated the proposed MRE on the *Fake News* dataset, which includes the claims and labeled tweets (the unknown tweets were not included). We calculated the minimal distance from each tweet to the given claim. A full demonstration of the proposed method is presented in the following example. For the given claim: *"Rihanna's Instagram message to followers to throw away the Snapchat app caused the company's share value to fall by hundreds of millions of dollars in one day."*. Tweet A: *"Rihanna Might Have Just Cost Snapchat $600 Million With a Single Instagram Story"*. Tweet B: *"Legends And Pop Stars As Social Media Lady Gaga Is Twitter Madonna Is Vine Rihanna Is Instagram Katy Perry Is Tumblr Cher Is Facebook Miley Cyrus Is Snapchat"*. Tweet A was labeled as *relevant*, whereas Tweet B as *irrelevant*.

We removed stop words and calculated the MRE for both tweets. Tweet A obtained an error of 0.948, as opposed to Tweet B, which obtained an error of 1.177. In Figure 2 it can be seen that according to the proposed distance-based approach, the words in Tweet A are closer to the words in the claim, than the words in Tweet B. For example, the words comprising the *irrelevant* Tweet B ("gaga", "miley", "starts") are far from the claim's words, in contrast to the words comprising the *relevant* Tweet A (e.g., "story" which is placed next to "message" or "millions" which is close to "hundreds"). Furthermore, we can see that the semantics of words according to the word embedding is preserved (e.g., the words that represent OSM platforms, such as "twitter", "facebook", "instagram", and "snapchat" are very close to each other). In general, we can conclude that the lower the MRE, the higher the probability of the retrieved tweet to be relevant to the given claim. We generated an area under the receiver operating characteristic curve (AUC) based on the *relevant* and *irrelevant* tweets. We used this measure since it enables us to examine different MRE thresholds values in contrast to other measures, such as precision, recall, and F1 which don't allow us to do this . As shown in Figure 3b, we achieved an AUC of 0.9 based on 1,078 labeled tweets related to 20 claims. Therefore, we can conclude that the proposed MRE is very useful for differentiating between *relevant* and *irrelevant* posts associated with a given claim.

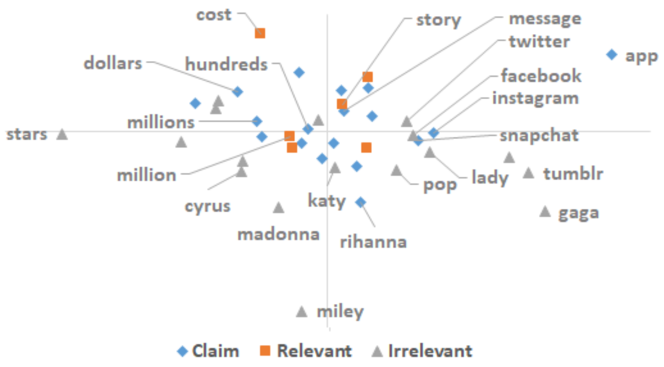


Fig. 2: A claim, relevant tweet, and irrelevant tweet embedded in 2D space.

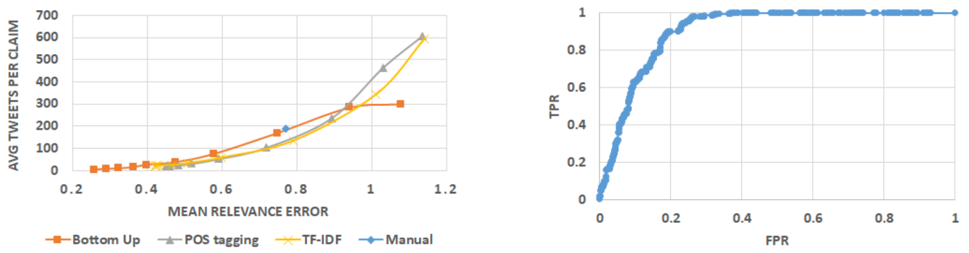
4.3 Keyword Optimization

In order to evaluate the proposed method, we compared it with three baseline methods for keyword selection: manual keyword assignment, TF-IDF, and POS tagging.

[4] http://commoncrawl.org/

[5] https://www.wikipedia.org/

[6] https://fasttext.cc/



(a) Average tweets per claim versus mean relevance error. The dots represent keywords that consist of ten to one words from left to right.

(b) ROC of our proposed measure on 1078 labeled posts.

Fig. 3: Evaluation

Algorithm 2 Manual Keyword Assignment

```
Read claim title and description
    Read the full report, if necessary.
Assign keywords that reflect the meaning of the claim.
Provide 3-5 alternative sets of keywords.
    Use synonyms.
Query the OSM using the various sets of keywords.
    Read a few of the retrieved posts.
    Assess the relevance of the retrieved posts.
    Record the number of retrieved posts.
```

Manual Keyword Assignment One of the methods used for collecting online discussions related to a given claim was a manual selection of high quality keywords. The guidelines for manual keywords selection follows.

First, read the given claim and understand its subject. Second, assign keywords, that reflect the meaning of the given claim. In addition, similar to [24], extract 3-5 keywords from the title and description of the claim. Third, expand the keywords using synonyms in order to retrieve posts that are written differently but convey the same message [20]. For example, for the claim: ”Did Donald Trump Scare a Group of Schoolchildren?”, there are several synonyms that can be used: Donald Trump - President of U.S., scare - frighten, schoolchildren - youngsters, etc. Four, verify the relevance of the assigned keywords using the search engine. Inspect the results and read a few of them in order to understand whether they are *relevant* to the given claim. Five, because the number of retrieved posts is important, obtain this value. In cases in which there are many posts, it recommended to use more then 3-5 synonyms as keywords (see Algorithm 2).

TF-IDF Keyword Generator The claims’ text was used as the targeted corpus. In addition, we removed stop words and computed the TF-IDF score for each word. Then, for each claim, we picked the K words with the highest TF-IDF score, where K is the number of required words.

POS Tagging Keyword Generator Here, we used part of speech (POS) tagging to generate keywords for each claim. Using this method, we narrow down the text to the following candidates: nouns, adjectives, adverbs, and numbers, based on the heuristics suggested by [12]. We prioritized the words by their POS tagging as follow:

$$other \leq number \leq adverb \leq adjective \leq noun$$

Then, the K first words from the candidates are selected as input keywords. Using this method, we prefer nouns over adjectives, adjectives over adverbs and so on. In a case in which two words with the same POS tag, we chose words based on their order of appearance in the document.

We generated TF-IDF and POS tagging keywords with a fixed size of one to ten words. The keywords defined manually were created using the manual keyword assignment guidelines. Then, we queried Twitter using the keywords and collected the top 600 posts and computed MRE on them. We can see that there is a trade-off between the number of posts retrieved per claim and their relevance. It is not surprising that longer queries are less beneficial than shorter queries due to the low number of results [20]. However, the proposed *Bottom Up* method outperforms the automated baseline methods (TF-IDF and POS tagging) and performed similarly to manual keyword selection (see Figure 3a). Moreover, on average, *Bottom Up* received more relevant posts then those received by TF-IDF and POS tagging.

5 Ethical Considerations

Collecting information from OSM has raised ethical concerns in recent years. To minimize the potential risks that may stem from such activities, this study follows recommendations presented by [5], which deal with ethical challenges regarding OSM and Internet communities.

In this study, given news article, we proposed a method that suggests the optimal keywords needed to retrieve the maximal number of relevant documents. To evaluate the proposed method, we used the Twitter search engine to retrieve tweets associated with the given news article. This service collects tweets published by accounts that agreed to share their information publicly. The research protocol was approved by the institutional Human Research Ethics Committee.

6 Conclusion & Future Work

In this paper, we proposed a novel automatic iterative approach for improving information retrieval from opaque search engines. We focused on the task of relevant posts retrieval from the Twitter OSM platform for a given news article. For this purpose, we proposed the *mean relevance error* MRE which estimates the relevance of results based on the mean distance between vector representations of the words in the article and words in the post. This estimation based on word embedding was found to be effective for distinguishing between *relevant* and *irrelevant* posts (see Section 4.2). The MRE can be helpful for automatically collecting relevant posts associated with a given claim. For example, the proposed *Bottom-Up* greedy algorithm attempts to construct a set of keywords by adding a keyword that improves the relevance of the retrieved posts in each iteration. The proposed *Bottom-Up* method was found to perform better than baseline methods, such as TF-IDF and POS tagging. Moreover, the performance of the automatic *Bottom-Up* method was very similar to the performance for manual keywords assignment (see Section 4.3).

We also presented the *Fake News* dataset (claims and tweets) used in our evaluation, as well as guidelines for manual labeling of tweets (see Section 4.1) and manual keyword assignment (see Section 4.3). Future work includes demonstrating the proposed approach on different OSM platforms, such as Reddit[7], Quora[8], etc. and the implementation of a framework that detects fake news based on OSM posts. Such a framework will use the ability of the proposed approach to extract automatic keywords on a large scale.

References

1. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 787–788. ACM (2007)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
3. Chirita, P.A., Firan, C.S., Nejdl, W.: Personalized query expansion for the web. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 7–14. ACM (2007)
4. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 299–306. ACM (2002)
5. Elovici, Y., Fire, M., Herzberg, A., Shulman, H.: Ethical considerations when employing fake identities in online social networks for research. Science and engineering ethics **20**(4), 1027–1043 (2014)
6. Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A.: Micro-blogging as online word of mouth branding. In: CHI'09 extended abstracts on human factors in computing systems. pp. 3859–3864. ACM (2009)

[7] https://www.reddit.com/

[8] https://www.quora.com/

7. Jurgen Koenemann, J., Belkin, N.J.: A case for interaction: A study of interactive information retrieval behavior and effectiveness. In: Proceeding of the ACM SIGCHI Conference on Human Factors in Computing Systems. pp. 205–212. Citeseer (1996)
8. Kenter, T., De Rijke, M.: Short text similarity with word embeddings. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 1411–1420. ACM (2015)
9. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International Conference on Machine Learning. pp. 957–966 (2015)
10. Kuzi, S., Shtok, A., Kurland, O.: Query expansion using word embeddings. In: Proceedings of the 25th ACM international on conference on information and knowledge management. pp. 1929–1932. ACM (2016)
11. Li, C., Wang, Y., Resnick, P., Mei, Q.: Req-rec: High recall retrieval with query pooling and interactive classification. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. pp. 163–172. ACM (2014)
12. Liu, P., Azimi, J., Zhang, R.: Automatic keywords generation for contextual advertising. In: Proceedings of the 23rd International Conference on World Wide Web. pp. 345–346. ACM (2014)
13. Liu, T., Ding, X., Chen, Y., Chen, H., Guo, M.: Predicting movie box-office revenues by exploiting large-scale social media content. Multimedia Tools and Applications **75**(3), 1509–1528 (2016)
14. Makki, R., Carvalho, E., Soto, A.J., Brooks, S., Oliveira, M.C.F.D., Milios, E., Minghim, R.: Atr-vis: Visual and interactive information retrieval for parliamentary discussions in twitter. ACM Transactions on Knowledge Discovery from Data (TKDD) **12**(1), 3 (2018)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
16. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)
17. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
18. Shu, K., Wang, S., Liu, H.: Understanding user profiles on social media for fake news detection. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). pp. 430–435. IEEE (2018)
19. Volkova, S., Shaffer, K., Jang, J.Y., Hodas, N.: Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 647–653 (2017)
20. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 61–69. Springer-Verlag New York, Inc. (1994)
21. Wang, J., Zhou, Y., Li, L., Hu, B., Hu, X.: Improving short text clustering performance with keyword expansion. In: The Sixth International Symposium on Neural Networks (ISNN 2009). pp. 291–298. Springer (2009)
22. Xu, B., Lin, H., Lin, Y., Yang, L., Xu, K.: Improving pseudo-relevance feedback with neural network-based word representations. IEEE Access **6**, 62152–62165 (2018)
23. Zamani, H., Dadashkarimi, J., Shakery, A., Croft, W.B.: Pseudo-relevance feedback based on matrix factorization. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 1483–1492. ACM (2016)
24. Zhang, C.: Automatic keyword extraction from documents using conditional random fields. Journal of Computational Information Systems **4**(3), 1169–1180 (2008)
25. Zhou, X., Zafarani, R., Shu, K., Liu, H.: Fake news: Fundamental theories, detection strategies and challenges. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 836–837. ACM (2019)