Iterative Query Selection for Opaque Search Engines with Pseudo Relevance Feedback

Aviad Elyashar, Maor Reuben, and Rami Puzis
Telekom Innovation Laboratories and Department of Software and Information Systems Engineering,
Ben-Gurion University of the Negev, Beer-Sheva, Israel
{aviade,maorreu}@post.bgu.ac.il, puzis@bgu.ac.il

Abstract

Retrieving information from an online search engine is the first and most important step in many data mining tasks. Most of the search engines currently available on the web, including all social media platforms, are black-boxes (a.k.a opaque) supporting short keyword queries. In these settings, retrieving all posts and comments discussing a particular news item automatically and at large scales is a challenging task. In this paper, we propose a method for generating short keyword queries given a prototype document. The proposed *iterative query selection* algorithm (IQS) interacts with the opaque search engine to iteratively improve the query. It is evaluated on the Twitter TREC Microblog 2012 and TREC-COVID 2019 datasets showing superior performance compared to state-of-the-art. IQS is applied to automatically collect a large-scale fake news dataset of about 70K true and fake news items. The dataset, publicly available for research, includes more than 22M accounts and 61M tweets in Twitter approved format. We demonstrate the usefulness of the dataset for fake news detection task achieving state-of-the-art performance.

Keywords: query selection, opaque search engine, pseudo relevance feedback, fake news

1. Introduction

Every day, millions of people search for information online (Christopher D. Manning & Schütze, 2008). Market researchers search for products related to their product or business (Yao et al., 2012). Researchers reviewing the academic literature search for works related to their current article (Bethard & Jurafsky, 2010). Posts and comments that discuss a news item are retrieved from online social media (OSM) for fake news detection (Zhou et al., 2015). There are many similar cases when additional information related to a specific document is required. We will refer to such a document as a *prototype*.

There are multiple methods for retrieving a set of documents that are similar to a given prototype from a corpus. Most of these methods represent documents as vectors and calculate the similarity between the prototype and other documents. The basic methods are based on TF-IDF (term frequency-inverse document frequency) and BM25 which treats each document as a bag-of-words Alvarez & Bast (2017). Advanced methods, such as Doc2Vec (Le & Mikolov, 2014), Skip-thoughts (Kiros et al., 2015), Sent2Vec (Pagliardini et al., 2018), and others, use neural networks to represent documents as low-dimensional vectors (Alvarez & Bast, 2017). Once vector representations of documents are available, retrieving the documents that are most similar to the prototype, i.e., closest to it in the embedding space, is straightforward.

Retrieval methods that are based on document similarity assume access to the corpus being searched. This assumption is valid for transparent search engines, where the repository and the algorithms are known to the user. However, all the popular search engines, including general-purpose search like Google, or platform-specific, such as Twitter search, are opaque providing very little information about their repositories and algorithms (urgen Koenemann & Belkin, 1996). Other than Google's image search, current search engines do not provide document-based search services. Therefore, we usually resort to the short keyword queries that are a mainstay of anyone using today's search engines (Chirita et al., 2007).

Due to the ambiguity of short keyword queries, they often do not reflect the original intention of the query writer (Cronen-Townsend et al., 2002). For example, the keyword "apple" may refer to the fruit or to the technology company.[1] In this paper, we focus on

[1] Google and some other engines use the search context to resolve ambiguity and retrieve documents that are most relevant for a specific user or case (Finkelstein et al., 2001). Other engines, such as Twitter, do

the problem of *retrieving documents that are most similar to a given prototype document from an opaque search engine supporting short keyword queries*, such as Twitter. It is possible to manually formulate a search query from the document's content (Zhou et al., 2015). But of course, manual query selection does not scale. One can use the prototype document's title to generate queries (Monti et al., 2019), or search for its URL if the prototype is a web page (Vosoughi et al., 2018). However, these approaches miss many relevant results. We further discuss the pros and cons of existing query selection methods in Section 2.

Therefore, in this paper, we suggest a novel iterative approach that selects queries that maximize the number of retrieved relevant results, using limited interaction with an opaque search engine. This approach consists of two components: the *iterative query selection (IQS)* algorithm and the *word's mover distance (WMD)* measure. The *IQS* is a hill climbing algorithm that iteratively optimizes short keyword queries given a prototype document. The *WMD* is used as pseudo relevance feedback, by ranking the results of incumbent queries generated by the IQS algorithm according to their relevance to the prototype document. The details of the proposed method are discussed in Section 3. In the absence of a prototype document, incumbent results are compared to a set of relevant results according to user relevance feedback. We evaluated the proposed methods on TREC 2012 Microblog benchmark and TREC-COVID 2019 datasets assuming relevance feedback (see Section 4) In addition, we utilized the proposed IQS algorithm to retrieve a large-scale fake news dataset from Twitter to be used toward training fake news classifiers (see Section 5).

The contributions of this paper are:

- automated mechanism for optimizing short keyword queries termed *iterative query selection (IQS)* (see Section 3.2). The IQS outperforms existing opaque relevance feedback search on Twitter TREC Microblog 2012 and on TREC-COVID 2019 datasets (see Section 4.3).

not use contextual information and retrieve all results exactly matching the specified keywords (Twitter). Personalization and context aware information retrieval are out of the scope of this paper.

- a large-scale *Fake News* dataset:[2] 70K news items discussed by 20M twitter users in 61M tweets (see Section 5.1).
- the quality of the dataset is demonstrated through the application of fake news detection algorithms on the collected data achieving AUC[3] of 0.92 and accuracy of 0.86 (see Section 5.4).

The rest of the paper is organized as follows: In Section 2, we review previous approaches for query selection, result diversification, document similarity, and fake news collection. In Section 3, we present the proposed iterative query selection algorithm for optimizing short keyword queries sent to an opaque search engine. In Section 4, we present the datasets used for evaluating the solution proposed, as well as discuss the results obtained. Section 6 discusses ethical considerations, and we conclude the paper in Section 7 with our plans for future work.

2. Related Work

This paper describes a new approach for selecting the best queries engaging with opaque search engines. In the next sections, we elaborate on the methods that are associated with the query selection. Next, for the demonstration of fake news detection use cases, we provide the necessary background for this domain.

2.1. Query Selection for Transparent Search Engines

Query selection is the task of selecting the most suitable queries for extracting relevant documents from web search engines (Wu et al., 2006). In most cases, selecting these queries requires reformulation or expansion of an initial query. Several methods suggest analyzing the underlying corpus of the given search engine and used this valuable information for expanding the queries. Roy et al. (2016) selected the most similar terms to a given query for expansion based on word embedding trained on the corpus. Their idea was to choose terms that yielded the highest probability of being

[2] The dataset is publicly available as a collection of Twitter IDs in the following link: https://bit.ly/2vd58u6

[3] Area Under the Receiver Operating Characteristic Curve

related to the current query. Kuzi et al. (2016) also used word embedding trained on the corpus to select expansion terms and suggested centroid- and fusion-based terms scoring methods to select them. Xu et al. (2018) selected candidate terms for expansion based on context features, such as TF-IDF and co-occurrence of the query terms. Afterward, they used the learned term-ranking models to rank the candidate terms. Pang & Du (2019) utilized click-through data of old queries for query reformulation. They first construct a click network that consists of queries and documents as nodes. Then they calculate the conditional probability of each term from the neighbor queries to be in the input query. Finally, they use the top terms to expand short queries and the tail terms to reduce long queries.

All these approaches require knowledge about the underlying corpus of the search engine and therefore are not suitable for the case of opaque search engines. Also, their starting point requires an initial query. Thus, it is not possible to use a prototype document in these approaches.

2.2. *Query Selection for Opaque Search Engines*

In opaque search engines, we lack knowledge about the underlying search method, corpus, or query selection method (if there is one). Thus, to optimize a query, we require an external query selection method. Such methods expand and reformulate an initial query using interactions with the search engine. Li et al. (2014) presented ReQ-ReC (ReQuery-ReClassify) a double-loop active retrieval system. The double-loop is a combination of an outer loop that is responsible for selecting new queries, and an inner loop that trains a document ranker using active learning. The process is finished when there are no more documents labeled as relevant from the user, or the user is satisfied with the results. ATR-Vis (Active Tweet Retrieval Visualization) is another retrieval system that was presented by Makki et al. (2018). This system is interactive and exploratory tool that detects tweets that are related to a given debate. To decrease user involvement in the process, ATR-Vis proposes four strategies of active learning. Ambiguous retrieval strategy sends tweets, that have a similar probability to relate to more than one debate, to labeling. In the near-duplicates strategy, tweets that have similar text, are labeled the same. The Leveraging hashtags strategy filters tweets containing hashtags that appear in multiple debates. In the leveraging replies strategy, tweets that their replies are classified uniformly among all debates are sent for labeling. Zamani et al. (2016) referred to the task of query expansion as a recommendation task. First, they consider the query and the retrieved pseudo-relevant documents as users, and the terms as items. Then they use non-negative matrix factorization to recommend terms for the given query. Another approach for query reformulation was introduced by Al-Khateeb et al. (2017), where the initial query can be reformulated using a genetic algorithm search. The synonyms of the query terms are candidates for the reformulation, and the fitness function is based on the similarity between the query and the results. Nogueira & Cho (2017) presented a neural network architecture that reformulates a query. The network receives the query terms and a given candidate term as input. Then, it predicts whether the candidate term is suitable for expanding the query. Chy et al. (2019) proposed a query expansion method that selects effective expansion terms using a random forest trained on term features. The extracted features are grouped into five categories: lexical features, Twitter-specific features, temporal features, sentiment features, and embedding-based features. ALMIK is an active retrieval method that tries to achieve both high-precision and high-recall in collecting event-related tweets (Zheng & Sun, 2019). Similar to the ReQ-ReC method, ALMIK contains a keyword expansion component, that improves the initial set of keywords iteratively, and an event-related tweet classifier that identifies related tweets. To reduce annotation effort, the tweet classifier is trained using a multiple-instance learning process. This process assigns labels to bags of similar instances.

These approaches, except for ATR-Vis, require an initial query. The drawback of the ATR-Vis is that it requires users to label the retrieved results (relevance feedback). In contrast, our proposed method uses a pseudo-relevance feedback process that does not require user interaction and can be used on any search engine.

2.3. Document Similarity

Over the past years, various solutions have been suggested for estimating the semantic similarity between documents based on lexical matching, handcrafted patterns, syntactic parse trees, external sources of structured semantic knowledge, and distributional semantics.

We can divide document similarity measures into two groups: the supervised measures and the unsupervised measures. The supervised measures require training to provide a similarity score for a pair of documents. Kenter & De Rijke (2015) generate multiple types of meta-features from texts' word embedding to train a supervised learning classifier. Later, they used the trained model for predicting the semantic similarity of new, unlabelled pairs of short texts. Deep relevance matching model (DRMM) (Guo et al., 2016) is a supervised model for determining the relevance of a document given a particular query. The proposed model employed a joint deep architecture at the query term level that estimates the query document similarity. Mitra et al. (2017) also suggested a supervised document ranking model composed of two separate deep neural networks, where the first network matches the query and the document using a local representation and the second network matches the query and the document using learned distributed representations. Next, the two networks are jointly trained as part of a single neural network. They showed that this combination performed better than either neural network individually on a web page ranking task and significantly outperformed traditional baselines and other recently proposed models based on neural networks.

The unsupervised document similarity measures provide a similarity score for a pair of texts without the requirement of training. The dual embedding space model (DESM) proposed by Nalisnick et al. (2016) calculates the average cosine distance of each term in the query with the centroids of the documents using pre-trained word embeddings. Another unsupervised document similarity measure is the word mover's distance (WMD) proposed by Kusner et al. (2015). It measures the dissimilarity between two documents as the minimal sum of distances that the word vectors of one document need to move towards the word vectors of another document. In this paper, we use WMD and extend it to a collection of retrieved documents.

2.4. Search Result Diversification

In many cases, queries for search engines can arguably be considered ambiguous to some extent. Therefore, in order to tackle query ambiguity, search result diversification approaches have recently been proposed to produce rankings for satisfying the multiple possible information needs underlying a query (Drosou & Pitoura, 2010). In most cases, the diversification of retrieved results implies a trade-off between having more relevant results that reflect the true intent of the user and having less redundancy (Gollapudi & Sharma, 2009). There are two prominent diversification approaches: implicit and explicit. The former approach implicitly assumes that similar documents will cover similar interpretations or aspects associated with the query and should hence be dismissed. In particular, an implicit representation of aspects relies on document features such as the terms contained in the retrieved documents (Carbonell & Goldstein, 1998), the clicks they received (Slivkins et al., 2010), their topic models (Carterette & Chandar, 2009), or clusters (He et al., 2011). The latter, explicit approach, allows a broad topic associated with an ambiguous query to be decomposed into its constituent sub-topics. Therefore, we can explicitly search for different aspects of the query for producing a diverse ranking of results. In most of the cases, explicit approaches rely on features derived from the query as candidate aspects, such as different query categories (Agrawal et al., 2009) or query reformulations (Santos et al., 2010).

In this paper, we diversify the returned documents using two query expansion methods: adding synonyms based on WordNet or Adding the k closest words in the embedding space for each candidate keyword in the query.

2.5. Fake News Data Collection Methods

Fake news is a long-lasting problem that has drawn significant attention in recent years. It has been widely spread within the online social media (OSM) (Willmore, 2016). Since fake news detection is very challenging, many researchers suggested different approaches to confronting this issue. Many of them were based on natural language processing (Zhou et al., 2019b; Zhou & Zafarani, 2020). Others were investigating the diffusion of news (Vosoughi et al., 2018; Zhou & Zafarani, 2020). Also, a few papers have attempted to detect fake news solely using social context features (Shu et al., 2017).

In order to train supervised classifiers for fake news detection, a ground truth dataset containing labeled news items is required. Such news items can be collected from fact checking websites, such as Snopes,[4] PolitiFact,[5] FactCheck,[6] and others (Vosoughi et al., 2018; Wang, 2017).

There are two commonly-used methods for collecting relevant posts associated with a given claim: The first method is to retrieve posts based on the sources that distributed the claims. For example, Monti et al. (2019) used the source's headlines that exist in fact-checking websites to collect tweets. Vosoughi et al. (2018) investigated the diffusion of news, based on collected tweets that contained links to the given claims. However, collecting tweets based on sources may be incomplete since many posts are associated with the given claim, but do not contain a link to the claim's source. Moreover, URL shortening, quotation, and cross-reference common in the press, as well as among bloggers, lead to a situation where tweets mentioning the same news contain links to different sources. Therefore, collecting tweets solely based on links will result in a subset of the tweets relevant to a claim. In addition, also the use of the source's headlines does not always reflect well the claim's content (e.g., in the case of clickbait) which can lead to irrelevant results. These drawbacks limit the ability to collect a quality dataset that contains enough relevant data for accurate classification.

The second method that people use to collect relevant posts is through the use of manual query selection. For example, Zhou et al. (2015) demonstrated a real-time news certification system on Sina Weibo[7] using queries provided by the user to gather related posts. Then, they built an ensemble model that combined user-based, propagation-based, and content-based features and evaluated the proposed model on a small dataset of 146 claims. Jin et al. (2017) and Wang et al. (2018) developed neural network-based methods for fake news detection and to evaluate their proposed methods they both used two small datasets from Sina Weibo (40k tweets) and Twitter (15k tweets on 52 rumor-related events). Those datasets were created using manual query selection. Selecting queries manually to a large collection of claims requires a lot of human effort and limits the amount of collected data.

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

[5] https://www.politifact.com/

[6] https://www.factcheck.org/

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

Due to the limitations of both methods described above, it is clear that fake news detection based on the OSM can benefit from a tool that can automatically select accurate short keyword queries for a given claim (i.e., a news item). In this study, we demonstrate the usefulness of the proposed *iterative query selection (IQS)* method to retrieve a large scale fake news dataset from Twitter automatically, as well as train fake news classifiers using social context features extracted from the tweets.

3. Iterative Query Selection with Word Mover's Distance Objective Function

In this paper, we propose a novel iterative approach for optimizing short keyword queries given a prototype document through interaction with an opaque search engine. First, we describe the *word mover's distance (WMD)*, a measure suggested by Kusner et al. (2015) that estimates the similarity of results to a given prototype document. This measure is calculated by summing the shortest distances between words in the given prototype document and words in the retrieved results (see Section 3.1). The lower the WMD, the more relevant the retrieved results are. Second, we outline the *iterative query selection (IQS)* algorithm for finding queries that retrieve results with the lowest *WMD* score (see Section 3.2).

3.1. Word Mover's Distance

In this section, we describe the word mover's distance (WMD) measures and its aggregation for multiple documents the mean word mover's distance (MMD).

The WMD measure estimates the minimal distance between word vector representations of the words existing in the retrieved result and the prototype document. The intuition is that documents that are close in their semantic space, probably discuss the same topic.

Let d denote the prototype document and r denote a short document retrieved using a search engine. Let w be a word vector representation that is calculated using a pre-traind word embedding method such as GloVe (Pennington et al., 2014), Word2Vec (Mikolov et al., 2013), FastText (Bojanowski et al., 2017), etc.

We can use any word embedding method, where words with a similar meaning are embedded close to each other. Let $dist(w_i, w_j)$ denotes the cosine distance between the vector representations of two words ($w_i, w_j \in \mathbb{R}^n$). The cosine distance is defined as $1 - cosineSim(w_i, w_j)$. Thus the cosine distance ranges between 0 to 2.

Let $W_d = \{w_{d_1}, w_{d_2}, \ldots, w_{d_l}\}$ be the set of word vectors in d and $W_r = \{w_{r_1}, w_{r_2}, \ldots, w_{r_k}\}$ be the set of word vectors in r. The W_d and W_r do not contain stop words. The distance between a word w and a document d is the minimal distance between the word w_i and all the words in W_d (see Equation 1).

$$dist(w_{r_i}, d) = \min_{w_{d_j} \in W_d} \{dist(w_{r_i}, w_{d_j})\} \tag{1}$$

The distance between the word vectors of the word w_{r_i} and all the words in W_d reflects the semantic similarity of the word w_{r_i} to the prototype document. The smaller the distance the higher semantic similarity.

Given a result document r, let word mover's distance (WMD) of r with respect to the prototype document d be the average distance of all words $w_{r_i} \in W_r$ to the document d (see Equation 2):

$$WMD(r, d) = \frac{1}{|W_r|} \sum_{w_{r_i} \in W_r} dist(w_{r_i}, W_d) \tag{2}$$

Where $|W_r|$ represents the number of words in W_r except stop words. It is important to mention that the stop word removal does not impact the rationality of the proposed method. However, mutual stop words in the result and the prototype documents do not indicate that both documents are similar semantically and thus discarded.

Note that, although the WMD is a binary function defined on pairs of documents, it is not a distance metric. The WMD is not symmetric and $WMD(r, d) = 0$ does not mean that r and d are equal in any sense. Rather WMD is similar to a fuzzy version of set inclusion ($\subseteq$), where $WMD(r, d) = 0 \implies W_r \subseteq W_d$. If r contains only words in d or their synonyms, $WMD(r, d)$ will be close to zero. The WMD works best when r is shorter than d since d may be covering multiple topics that are not mentioned in r.

In the final step, we set the mean word mover's distance (MMD) measure to estimate the similarity of the multiple results to a given prototype document. Let R be a set of short documents retrieved from a search engine. We define the MMD as the mean

WMD of all results $r \in R$ with respect to the prototype d (see Equation 3):

$$MMD(R, d) = \frac{1}{|R|} \sum_{r \in R} WMD(r, d) \quad (3)$$

The MMD defined above is designed to measure only one aspect of query performance, the relevance of the results. Other important aspects, for example, the number of results, are intentionally not captured by the MMD. The quality of the MMD is affected by the quality of the underlying word embedding model. For general purpose query evaluation, it is recommended to use word embedding models trained on large non-domain specific datasets.

3.2. Iterative Query Selection

The proposed *iterative query selection (IQS)* method is based on a local search algorithm, which selects the queries that maximize the relevance of the corresponding results retrieved from an opaque search engine. We use the hill climbing algorithm since querying the search engine is resource-intensive and we need to find local optimum with only a few iterations (Skiena, 2020).

Let d be a prototype document and W_d be the set of words in d as in the previous subsection. W_d does not contain stop words. In addition, named entities, e.g., "Michael Jordan," are considered as a single term if they are found in the vocabulary of the word embedding approach used as the basis for the WMD.

Let V_d denote the vocabulary of terms from which possible queries $q \in V_d$ are selected. V_d may be equal to W_d or expanded using any query expansion approach. We consider two query expansion methods: (1) Adding synonyms based on WordNet Miller (1995) for each word in W_d, later referred to as *Syn*. (2) Adding k closest words in the embedding space for each candidate word in W_d, later referred to as *KNN*.

The IQS searches through the space of possible queries $q \in V_d$. It starts with a random subset of words from V_d. For efficiency, the query size is limited by two control variables $minq$, and $maxq$, which are the minimal and the maximal number of words in a query. In every iteration, we randomly modify the query using one of the following three actions: $\textsc{AddWord}(q, V_d)$ randomly adds to q a word from V_d that is not yet in q. $\textsc{RemoveWord}(q, V_d)$ removes a random word from the query q decreasing its size.

SwapWords(q, V_d) exchanges a random word in q with a random word in V_d that was not already in q. Possible actions are chosen to ensure the query size constraints.

After modifying the query q using one of the three actions, we evaluate the MMD of its results R_q from the search engine se. Due to computational and network performance considerations, it is important to limit the number of results retrieved from se in each iteration of the algorithm. Usually, this limit further referred to as $rlimit \geq |R_q|$, is defined by the search engine interface and is set to the number of results on a single page. The larger the $rlimit$ is, the more accurate the $MMD(R_q, d)$ since it will be calculated on more retrieved documents. However, the $rlimit$ is also the primary factor (linearly) affecting the time of an iteration.

The hill climbing IQS algorithm is implemented as described in Algorithm 1. It receives as an input a prototype document d, an opaque search engine se, the maximal and minimal number of words in a query ($maxq$ and $minq$, respectively), the maximal number of iterations itr, and the number of result documents ($rlimit$) retrieved from se in each iteration. During the algorithm, we keep only queries that decrease the MMD score. If the query returns no results, we mark the query as irrelevant by setting its $MMD(R_q, d)$ score to be the maximal score of 2.

The IQS returns an ordered set of queries. Some search engines allow words from the query to be missing in the results while others retrieve only documents containing all keywords in the query. Twitter is an example of the latter, a boolean search engine. In the case of a boolean search engine, it is important to run multiple slightly modified queries in order to retrieve as many relevant results as possible. This is the main reason due to which the IQS returns a list of queries and not only a single best query.

4. Experiments

In this section, we describe a series of experiments that evaluate our iterative query selection (IQS) method. Since IQS required a loss function that evaluates each generated query, we first measure the ability of the WMD (word mover's distance) measure to distinguish between relevant and irrelevant results (see Section 4.2). Then we examine the WMD performance correlation to the informativeness of the prototype document.

Algorithm 1 Iterative Query Selection

```
procedure BuildQueries($d, se, itr, minq, maxq, rlimit$)
    $queries \leftarrow$ empty priority queue
    $V_d \leftarrow$ filtered and expanded set of words in $d$
    $q_{best} \leftarrow$ random subset of $V_d$
    $R_{q_{best}} \leftarrow se(q_{best}, rlimit)$
    calculate $MMD(R_{q_{best}}, d)$
    $q_{new} \leftarrow q_{best}$
    $R_{q_{new}} \leftarrow R_{q_{best}}$
    loop $itr$ times
        $actions = \{$AddWord, RemoveWord, SwapWords$\}$
        if $|q_{new}| = maxq \vee |R_{q_{new}}| = 0$ then remove AddWord from $actions$
        else if $|q| = minq$ then remove RemoveWord from $actions$
        $action \leftarrow random(actions)$
        $q_{new} \leftarrow action(q_{best}, V_d)$
        $R_{q_{new}} \leftarrow se(q_{new}, rlimit)$
        Using Eq.(3), calculate $MMD(R_{q_{new}}, d)$
        if $MMD(R_{q_{new}}, d) < MMD(R_{q_{best}}, d)$ then
            $queries.add(q_{new}, MMD(R_{q_{new}}, d))$
            $q_{best} \leftarrow q_{new}$
    return $queries$
procedure AddWord($q, V_d$) return $q \cup random(V_d \setminus q)$
procedure RemoveWord($q, V_d$) return $q \setminus random(q)$
procedure SwapWords($q, V_d$) return RemoveWord(AddWord($q, V_d$), $V_d$)
```

After evaluating the WMD, we examine the performance of the IQS using the MMD (mean WMD) as an active retrieval method for an opaque search engine (see Section 4.3).

4.1. Datasets

In the following experiments, we used the *Twitter TREC Microblog 2012* dataset and the *TREC-COVID 2019* dataset. The *Twitter TREC Microblog 2012* consists of 59 topics (used as initial queries) and 73K judgments (relevant and irrelevant tweets) for those topics (Soboroff et al., 2012). The corpus was collected over two weeks, from January 23, 2011, to February 7, 2011, containing 16M tweets. The *TREC-COVID 2019* [8] consists of 35 topics and 20.7K judgments. It was collected from COVID-19 Open Research Dataset (CORD-19) [9] that contains biomedical articles related to COVID-19. This dataset was constructed to develop solutions that improve searching for reliable information on the virus and its impact.

4.2. The Word Mover's Distance (WMD) Evaluation

The WMD's purpose is to rank documents by their relevance to a prototype document. However, The Twitter TREC Microblog 2012 dataset and TREC-COVID 2019 dataset include topic definitions that cannot be used as prototype documents (initial query), due to their rather short length. Therefore, we iteratively construct such prototype documents for each topic using a relevance feedback process.

4.2.1. Experimental Setup

The constructed prototype document should reflect the topic being searched. We use the following general process to iteratively build a prototype document for each topic and improve the results retrieved using the WMD. First, we use the initial query for each topic as the prototype document. We calculate the WMD between the prototype document and each tweet in the dataset. Although the prototype document is too short, some relevant tweets can be found using the WMD. Second, we retrieve the top k results and request relevance feedback from a user (or from an oracle if ground truth is provided

[8] https://www.kaggle.com/c/trec-covid-information-retrieval/overview

[9] https://www.semanticscholar.org/cord19

for evaluation purposes). Next, we expand the prototype document using the content of the relevant retrieved results and run the second step again.

It is important to note that the relevance feedback should be saved to avoid labeling the same result multiple times. The process stops after n labeled results for each topic in the dataset (or a user is satisfied with the results). In this experiment, we set the top k results to 10 and n to be 300. For the TREC Microblog 2012 dataset, we discard query 76, since it does not contain any judgments labeled as relevant.

As baselines we use the following: Okapi BM25 (Robertson & Zaragoza, 2009), latent semantic analysis (LSA) (Deerwester et al., 1990), and TF-IDF. Also, we use the dual embedding space model (DESM) using the same pre-trained word embeddings used for the WMD (Nalisnick et al., 2016). In this comparison, we use only unsupervised document similarity measures since our proposed method should run on any search engine without training. Note that MB25, LSA, and TF-IDF are not purely unsupervised measures since they require knowledge of the corpus which can be argued as training. Since we have knowledge of the corpus in this experiment, we consider them unsupervised.

4.2.2. Results & Discussion

The mean average precision (MAP) and R-precision results are summarized in Table 1. As can be seen, the WMD outperforms other methods evaluated on both datasets in terms of MAP and R-precision. These results emphasize the effectiveness of the WMD in being a good indicator for distinguishing between relevant and irrelevant documents. Also, it strengths the pre-trained word vectors of being very useful to detect similar words in two documents.

Another impotent aspect we want to examine is the effect of the prototype document informativeness (number of labels) on the MAP score. This aspect is impotent because it examines whether the relevance measure estimates the results' relevance according to the prototype or not. The results are presented in Figure 1. The trends in the results show how the WMD utilizes better the information found in the prototype document to rank the results. This finding indicates that the WMD is the best candidate as a loss function for our query selection method.

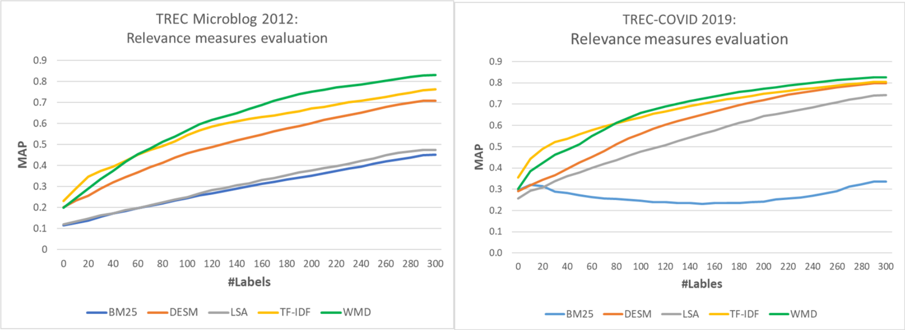

Figure 1: Evaluate the relevance measures performance relative to the number of active retrieval labels on the Twitter TREC Microblog 2012 and TREC-COVID 2019.

	TREC Microblog 2012		**TREC-COVID 2019**	
Method	**MAP**	**R_Precision**	**MAP**	**R_Precision**
BM25	0.451	0.384	0.335	0.345
LSA	0.474	0.410	0.741	0.676
DESM	0.707	0.639	0.798	0.749
TF-IDF	0.763	0.694	0.805	0.759
WMD	**0.831**	**0.780**	**0.825**	**0.788**

Table 1: Evaluation of the relevance measures on the Twitter TREC Microblog 2012 dataset and TREC-COVID 2019 dataset.

4.3. Iterative Query Selection with Relevance Feedback

In this experiment, we evaluate the full *iterative query selection (IQS)* pipeline based on a process that mimics interaction with Twitter's search engine. Twitter uses a boolean retrieval model, which means that the returned results must contain all the words in the query. We assume an opaque search engine (se) and access the corpus through the boolean search process, like in Twitter.

4.3.1. Experimental Setup

Similar to the previous experiment, we dynamically construct a prototype document for each topic using relevance feedback. First, we use the topic definition as the prototype document. Second, we run the first iteration of the IQS calculating the MMD (mean

WMD) score between the prototype document and the results retrieved. Third, before proceeding to the next iteration of the IQS, we sort the retrieved results in ascending order, according to their MMD score. Fourth, we take the top $k = 10$ results and ask a user (or an oracle) to label them. Fifth, we expand the prototype document using the content of the relevant results identified by the user (or the oracle) and proceed to the next iteration of IQS. The stopping condition is the same as in the previous experiment, retrieving labeling $n = 300$ tweets. We set the minimal and maximal number of words in a query to be between 1 to 6 ($minq = 1, maxq = 6$). This parameter influences the number of retrieved results directly. When a query containing a single word, many results are expected to be retrieved that probably most of them are not relevant directly to the prototype document. For example, assume that the prototype document is "Crude oil production in the U.S." and the query contains the word "oil", solely. In this case, the Twitter search engine is expected to retrieve many tweets that include the word although they are not related directly to the oil industry in the U.S. (for example, tweets that focus on oil painting and oil production in Russia). In the same manner, a high number of words decrease the number of retrieved results but most of these tweets are expected to be relevant to the given prototype document. Lastly, we set the number of returned results to 20 ($rlimit = 20$) to simulate a similar number of retrieved documents from a standard search engine within the Web.

In order to choose the best hyper-parameters for IQS, we tested several ranges: itr between 10 to 45, $runs$ between 1 to 3, and $numQueries$ between 5 to 50. For our final evaluation, we used the hyper-parameter configuration that yielded the best results on both datasets. During the hyper-parameter tuning, we limited the total number of interactions with the given search engine ($runs * itr$) to be up to 45, due to time constraints. Each search interaction with the Twitter search engine takes approximately 1.4 seconds. Therefore, the total run time for a claim is about a minute. Eventually, the best parameters found for IQS were $itr = 15, runs = 3, minSize = 1, maxSize = 6$, and $numQueries = 40$.

	TREC Microblog 2012		TREC-COVID 2019	
Method	**MAP**	**R_Precision**	**MAP**	**R_Precision**
ReQ-ReC	0.147	0.198	0.002	0.014
ALMIK	0.164	0.172	0.288	0.336
IQS	**0.357**	**0.356**	**0.508**	**0.507**

Table 2: Active retrieval methods comparison on the TREC Microblog 2012 and TREC-COVID 2019 datasets.

4.3.2. Results & Discussion

We compared the performance of the IQS to the ReQ-ReC implementation on GitHub[10] with the same settings: the top 10 documents are labeled by the user ($k = 10$), the algorithm stops after 300 labels for each topic ($n = 300$), and a boolean search engine.

In addition, we run the ALMIK method, a state-of-the-art active retrieval method proposed by Zheng & Sun (2019), on the TREC Microblog 2012 and TREC-COVID 2019 datasets. We implemented the ALMIK method based on the method description presented in their paper. Again, we limit the number of label requests from the user to 300 and use the same search engine mechanism. We also conducted hyper-parameter tuning for ALMIK in order to achieve the best results. The best results were achieved when the ALMIK conduct 3 rounds of active learning phases. Between the phases, we conduct a keyword expansion phase and used the new results in the next round. In each active learning phase, we conduct 10 iterations of 10 label requests of the most uncertain tweets from the user (A total of 100 in each active learning phase).

We reported the performance of the IQS, the ReQ-ReC, and the ALMIK all topics in both datasets Table 2. The proposed IQS method outperformed the ReQ-ReC and ALMIK in terms of MAP and R-Precision in both datasets. The results show that the IQS can retrieve more relevant results from a boolean opaque search engine using relevance feedback given a short initial query.

[10] https://github.com/lookatmoon/ReQ-ReC-demo

4.4. Iterative Query Selection with Keyword Expansion

In the above experiments, we used the *vanilla* IQS without any additional query expansion methods. Alongside the vanilla configuration, we also added two keyword expansion methods associated with IQS. In the first method, we expand each candidate word using its top five synonyms from WordNet (IQS+Syn). In the second method, each candidate word is expended using its five nearest neighbors words based on *FastText* word embedding (IQS+KNN).

4.4.1. Experimental Setup

We evaluated the performance of the three configurations (IQS, IQS+Syn, and IQS+KNN) with respect to the number of iterations, as follows: First, we executed each configuration with 85 iterations to examine where each configuration converges. In addition, we executed each configuration 3 times and presented the average performance. After every 5 iterations, we measured the MAP of the 5 queries selected. There are cases in which the query with the lowest score retrieves irrelevant documents. To avoid these cases, we used the top 5 queries to smooth the variation in performance. Figure 2 presents the performance of the configurations where each sub-figure reflects the number of labels retrieved using the active retrieval process.

4.4.2. Results & Discussion

As can be seen in Figure 2, vanilla IQS is found superior until a turning point where the IQS+KNN and the IQS+Syn outperformed the vanilla configuration. In addition, we see that vanilla IQS converged faster than the other configurations. It is expected since the vanilla IQS has a smaller search space than IQS+KNN and IQS+Syn configurations i.e., the algorithm requires fewer search iterations to converge. We can also see that the active retrieval iterations affect the performance of the configurations. The more active retrieval iterations the sooner the turning point arrives. The reason is that the ratio between the number of original keywords to the number of expended keywords decreases when the prototype is more informative. A low ratio means that there is a higher likelihood of using more original keywords and thus perform similarly to the vanilla configuration at the start. Since we want to keep a low number of interactions with the search engine, vanilla IQS is the best option.

5. Application for Fake News Detection with Pseudo Relevance Feedback

In many cases, it is not practical and scalable to ask for feedback from the users continuously. Therefore, here, we are in a mode of pseudo relevance feedback. This means that we apply the *iterative query selection (IQS)* with the mean word mover's distance (MMD) as pseudo relevance feedback. To reduce the number of interactions with the Twitter search engine we use the IQS vanilla configuration.

In this section, we demonstrate the importance of the IQS as a necessary link in the pipeline of fake news detection. Using the proposed IQS and MMD, we demonstrate an automated collection of relevant tweets associated with labeled news items (ground truth). Later, using these relevant tweets, we demonstrate fake news detection using supervised machine learning classifiers. First, we describe the background of fake news detection on online social media (OSM), including the data collection process. Afterward, we present the dataset obtained using the IQS for fake news detection on OSM. Finally, we train a classifier on the collected data and present its performance.

5.1. Data Collection With IQS

In this section, we describe the construction of a large dataset for the task of fake news detection on OSM using the *iterative query selection (IQS)*. First, we crawl news items from fact-checking websites, such as Snopes, Gossip Cop,[11] and Politifact. For Snopes and Politifact news items, there are five fine-grained multiple labels: true, mostly true, false, mostly-false and pants-on-fire. Similar to Rasool et al. (2019), we converted the classification problem into binary by categorizing the news items that their label are true and mostly-true as true and news items that their label are false, mostly-false and pants-on-fire as false. For Gossip Cop, we categorized news items with scores between 0 to 3 as false and news items with scores between 7 to 10, as true. Since the majority of the news items in fact-checking websites are false, we added news items from 10

[11] https://www.gossipcop.com/

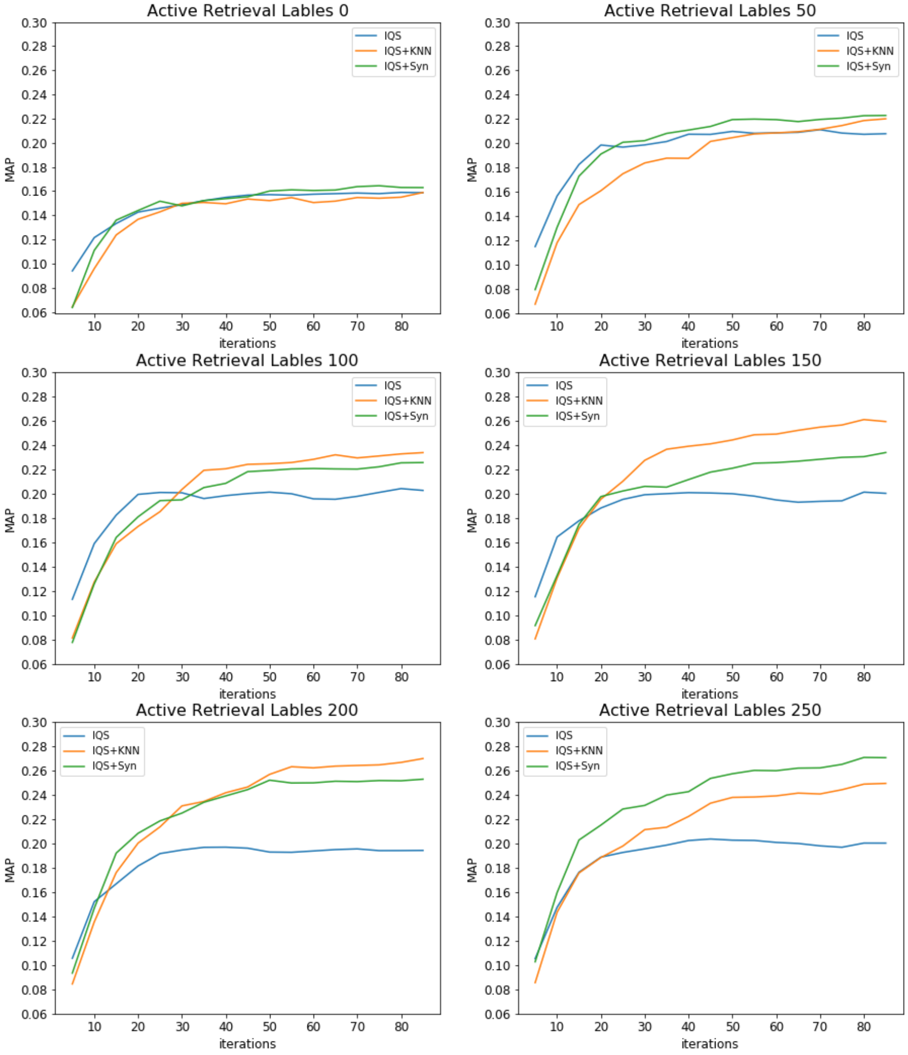

Figure 2: Performance evaluation of the IQS configurations on the Twitter TREC Microblog 2012. Each graph presents the performance after 0, 50, 100, 150, 200, and 250 active retrieval labels. The MAP score is calculated based on the top five selected queries for every five hill climbing iterations.

well-known news sources (Time of Israel, CNN News, ABC News, BBC News, The New York Times, The Jerusalem Post, The American Conservative, MSNBC, Fox News, and Politico) as true news items. A large number of studies exploited reliable news sources as a proxy for true news items (Monti et al., 2019). In total, we collected 70,018 news items (16,212 false, 53,806 true). For each news item, we set the IQS algorithm to run on Twitter API three times, with the following parameters: Returning 5 final queries ($numQueries = 5$, a maximal number of 15 iterations ($itr = 15$), five queries, the number of words in a query is between 3 to 6 ($minq = 3, maxq = 6$), and the number of results returned equals 20 ($rlimit = 20$). Then, after obtaining the top five final queries from all three runs, we use them in order to retrieve the most relevant tweets for each news item, while limiting the number of tweets returned for each query to 500. To make the fake news detection more realistic, we collected only the tweets that were posted before the fact-checker assigned a label for the given news item. Finally, utilizing this approach, we constructed a large fake news dataset containing about 70,000 news items and about 61 million corresponding posts, the distribution of which is shown in Table 3.

5.2. Fake News Classification

The task of fake news classification has been studied intensely in recent years. Along with the growth of online news, many non-traditional news sources, such as blogs, have evolved in order to respond to users' “appetite for information." In many cases, however, these sources are operated by amateurs whose reporting is often subjective, misleading, or unreliable (Downie & Schudson, 2009). This "everyone is a journalist" phenomenon (Zeng, 2018), coupled with the flood of unverified news and the absence of quality control procedures to prevent potential deception, has contributed to an increasing problem of fake news dissemination (Conroy et al., 2015).

The spread of misinformation, propaganda, and fabricated news has potentially harmful effects, even including a significant impact on real-world events (Allcott & Gentzkow, 2017). In recent years, it has weakened public trust in democratic governments and their activities, such as the “Brexit" referendum and the 2016 U.S. election (Zhou et al., 2019a). World economies are also not immune to the impact of fake

Domain	#News Items	#True	#False	#Authors	#Tweets
politifact.com	12,952	5,288	7,664	4,274,688	12,525,467
snopes.com	4,682	845	3,837	1,721,162	3,617,550
gossipcop.com	4,764	53	4,711	863,906	2,302,245
times of israel	21,989	21,989	0	5,401,027	20,963,687
Jerusalem Post	3,109	3,109	0	1,469,181	3,386,702
New York Times	3,453	3,453	0	1,641,931	3,247,247
abc-news	3,685	3,685	0	1,042,084	2,410,832
cnn-news	3,496	3,496	0	1,381,585	2,988,373
fox-news	3,620	3,620	0	1,115,282	2,921,356
bbc-news	3,687	3,687	0	1,181,490	2,469,421
msnbc-news	2,410	2,410	0	731,839	1,982,974
politico	1,640	1,640	0	853,716	1,844,441
American Conservative	531	531	0	440,086	619,336
Total	**70,018**	**53,806**	**16,212**	**22,117,977**	**61,279,631**

Table 3: Fake news dataset statistics.

news; this was demonstrated when a false claim regarding an injury to President Obama caused the stock markets to plunge (dropping 130 billion dollars) (Rapoza, 2017). In recent years, due to the threats to democracy, journalistic integrity, and economies, researchers have been motivated to develop solutions for this serious problem (Zhou et al., 2019a) proposing approaches for the detection of fake news based on natural language processing (Zhou et al., 2019b), investigating the diffusion of news (Vosoughi et al., 2018), etc.

5.3. Fake News Classification Method

To classify the news items, we extract author- and post-based features. For author-based features, we applied aggregation functions on various aspects of author demographics, such as registration age, number of followers, number of followees, number of distributed tweets published by the user, etc. Post-based features include the aggrega-

No.	Feature Name	Gini Importance
1	Number of verified authors	0.162
2	Max number of posts published by claim's authors	0.049
3	Glove_wikipedia_model_300d max dimension 295	0.046
4	Glove_wikipedia_model_300d min dimension 291	0.037
5	Max favorites count of claim's authors	0.036
6	Max followers count of claim's authors	0.033

Table 4: Features listed by Gini importance.

tions of posts metadata, such as retweet count, text length, the time interval between the oldest and newest post, etc. For all the features extracted from the post, we removed stop words. Also, regarding the post's data, we extracted the following features: sentiment, temporal (post diffusion patterns), LDA (variations on the posts' topics), TF-IDF, and word embedding. For the latter, we used the Glove Wikipedia pre-trained model with 300 dimensions. For aggregation functions, we used mean, median, max, min, standard deviation, kurtosis, and skewness functions.

5.4. Fake News Classification Results

For the classification, we tried many combinations of supervised machine learning algorithms and feature subsets. All classifiers were trained using 10-fold cross-validation. Eventually, we averaged the results obtained from all the folds. We determined that the best performing classifier on the test set was the Random Forest with 100 estimators and a max depth of 10. This classifier, with 100 features obtained AUC and accuracy of 0.92 and 0.86, respectively. **This result is evidence that an application that detects false news based on the OSM can benefit from training on data collected using IQS.**

Also, we analyzed most of the influential features of the best classifier (see Table 4). The most important feature was the number of verified authors with a Gini importance of 0.162 (see Table 4). Comparing the distribution of these authors with respect to fake and true news items, we can see the number of verified authors within true news is 3 times higher than in false news items. These differences were found statically significant (a p-value of 0.0). Based on this result, we conclude that verified authors are important actors for fake news detection. The higher their participation, the higher the reliability of the online discussion.

According to the Gini importance, the second, fifth, and sixth highest influential features were aggregations over the news item's authors. This strengthens the conclusion of Castillo et al. (2011) that author-based features are very relevant for fake news detection within the OSM.

In addition, the third and fourth features are aggregations on the word embedding of the news item's posts. These features indicate that the fixed-length of vector representations of the words consists of the online discussions that can hold the truthfulness of given news items.

These results show that our proposed algorithm can be utilized for solving real-world problems (e.g., the detection of fake news). In addition, the machine learning classifiers trained on the collected this large dataset using the IQS obtained impressive results. This strengthens that our method can be very useful for detecting fake news while retrieving relevant data automatically.

6. Ethical Considerations

Collecting information from OSM has raised ethical concerns in recent years. To minimize the potential risks of such activities, this study follows recommendations presented by (Elovici et al., 2014), which deal with the ethical challenges of OSM and Internet communities.

For this study, we proposed a method, that selects queries for a given prototype document to retrieve the maximal number of relevant documents. To evaluate the proposed method, we used the Twitter search engine in order to retrieve tweets associated with the given prototype document. This service collects tweets published by accounts that agreed to share their information publicly.

7. Conclusion & Future Work

In this paper, we propose an automated *iterative query selection (IQS)* algorithm for improving information retrieval from opaque search engines. This method consists of two components: the *mean word mover's distance (MMD)* which estimates the semantic similarity between the retrieved documents to the given prototype document and the iterative algorithm which selects suitable queries based on the mean WMD (MMD).

We evaluated IQS on the *Twitter TREC Microblog 2012* and *TREC-COVID 2019* datasets. The proposed IQS algorithm was found superior to the other two state-of-the-art methods on both datasets. Next, we applied IQS for obtaining a large fake news dataset that later was used for the task of fake news detection.

As a result, we conclude the following:

First, the WMD score is found to be a successful measure for differentiating between relevant and irrelevant documents concerning a given prototype document (see section 4.2). This result strengthens previous conclusions of Kusner et al. (2015) related to WMD being effective for document classification.

Second, the IQS algorithm obtained the highest performance with respect to two state-of-the-art methods: ReQ-ReC (Liu et al., 2014) and ALMIK (Chy et al., 2019) and found effective for active retrieval task.

Third, it is recommended to use the proposed IQS as part of an automated fake news detection pipeline. Using this algorithm, we collected a large-scale fake news dataset consisting of 70K news items, 22M accounts, and 61M tweets, automatically. Obtaining an AUC of 0.92 and an accuracy of 0.86 using classic machine learning classifiers emphasizes the quality of the large dataset collected using IQS.

One possible direction for future work could be to demonstrate the proposed approach on different OSM platforms, such as Reddit,[12] Quora,[13] etc. Another might compare the effectiveness of a fake news detection system using data collected using a source URL versus data collected using our query selection method.

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

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

8. Availability

This study is reproducible research. Therefore, the Fake News datasets is available.[14] Other datasets for evaluation are available upon request.

9. References

Agrawal, R., Gollapudi, S., Halverson, A., & Ieong, S. (2009). Diversifying search results. In *Proceedings of the second ACM international conference on web search and data mining* (pp. 5–14).

Al-Khateeb, B., Al-Kubaisi, A. J., & Al-Janabi, S. T. (2017). Query reformulation using wordnet and genetic algorithm. In *2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT)* (pp. 91–96). IEEE.

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, *31*, 211–36.

Alvarez, J. E., & Bast, H. (2017). A review of word embedding and document similarity algorithms applied to academic text. *bachelor thesis*, .

Bethard, S., & Jurafsky, D. (2010). Who should i cite: learning literature search models from citation behavior. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 609–618).

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Carbonell, J., & Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 335–336).

[14] https://drive.google.com/drive/folders/1nOGYjGoZHxFwaPm0xci-T7V0a90YW448?usp=sharing

Carterette, B., & Chandar, P. (2009). Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 1287–1296).

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web* (pp. 675–684). ACM.

Chirita, P.-A., Firan, C. S., & Nejdl, W. (2007). 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.

Christopher D. Manning, P. R., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Chy, A. N., Ullah, M. Z., & Aono, M. (2019). Query expansion for microblog retrieval focusing on an ensemble of features. *Journal of Information Processing*, *27*, 61–76.

Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community* (p. 82). American Society for Information Science.

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 299–306). ACM.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*, 391–407.

Downie, L., & Schudson, M. (2009). The reconstruction of american journalism. *Columbia Journalism Review*, *19*.

Drosou, M., & Pitoura, E. (2010). Search result diversification. *ACM SIGMOD Record*, *39*, 41–47.

Elovici, Y., Fire, M., Herzberg, A., & Shulman, H. (2014). Ethical considerations when employing fake identities in online social networks for research. *Science and engineering ethics*, *20*, 1027–1043.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web* (pp. 406–414).

Gollapudi, S., & Sharma, A. (2009). An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web* (pp. 381–390).

Guo, J., Fan, Y., Ai, Q., & Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 55–64).

He, J., Meij, E., & de Rijke, M. (2011). Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology*, *62*, 550–571.

Jin, Z., Cao, J., Guo, H., Zhang, Y., & Luo, J. (2017). Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 25th ACM international conference on Multimedia* (pp. 795–816). ACM.

Kenter, T., & De Rijke, M. (2015). Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (pp. 1411–1420). ACM.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294–3302).

urgen Koenemann, J., & Belkin, N. J. (1996). 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.

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning* (pp. 957–966).

Kuzi, S., Shtok, A., & Kurland, O. (2016). Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 1929–1932). ACM.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196).

Li, C., Wang, Y., Resnick, P., & Mei, Q. (2014). 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.

Liu, P., Azimi, J., & Zhang, R. (2014). Automatic keywords generation for contextual advertising. In *Proceedings of the 23rd International Conference on World Wide Web* (pp. 345–346). ACM.

Makki, R., Carvalho, E., Soto, A. J., Brooks, S., Oliveira, M. C. F. D., Milios, E., & Minghim, R. (2018). Atr-vis: Visual and interactive information retrieval for parliamentary discussions in twitter. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*, 3.

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

Miller, G. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, *38*, 39–41.

Mitra, B., Diaz, F., & Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 1291–1299).

Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*, .

Nalisnick, E., Mitra, B., Craswell, N., & Caruana, R. (2016). Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web* (pp. 83–84).

Nogueira, R., & Cho, K. (2017). Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 574–583).

Pagliardini, M., Gupta, P., & Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 528–540).

Pang, W., & Du, J. (2019). Query expansion and query fuzzy with large-scale click-through data for microblog retrieval. *International Journal of Machine Learning and Computing*, *9*.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Rapoza, K. (2017). Can 'fake news' impact the stock market? www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/.

Rasool, T., Butt, W. H., Shaukat, A., & Akram, M. U. (2019). Multi-label fake news detection using multi-layered supervised learning. In *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering* (pp. 73–77).

Robertson, S., & Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Roy, D., Paul, D., Mitra, M., & Garain, U. (2016). Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*, .

Santos, R. L., Macdonald, C., & Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web* (pp. 881–890).

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*, 22–36.

Skiena, S. S. (2020). *The algorithm design manual*. Springer International Publishing.

Slivkins, A., Radlinski, F., & Gollapudi, S. (2010). Learning optimally diverse rankings over large document collections. In *ICML*.

Soboroff, I., Ounis, I., Macdonald, C., & Lin, J. J. (2012). Overview of the trec-2012 microblog track. In *TREC* (p. 20). Citeseer volume 2012.

Twitter (). Search tweets. https://developer.twitter.com/en/docs/tweets/search/guides/standard-operators.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, *359*, 1146–1151.

Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 422–426).

Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., & Gao, J. (2018). Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 849–857). ACM.

Willmore, A. (2016). This analysis shows how viral fake election news stories outperformed real news on facebook.

Wu, P., Wen, J.-R., Liu, H., & Ma, W.-Y. (2006). Query selection techniques for efficient crawling of structured web sources. In *22nd International Conference on Data Engineering (ICDE'06)* (pp. 47–47). IEEE.

Xu, B., Lin, H., Lin, Y., Yang, L., & Xu, K. (2018). Improving pseudo-relevance feedback with neural network-based word representations. *IEEE Access*, *6*, 62152–62165.

Yao, J., Yao, J., Yang, R., & Chen, Z. (2012). Product recommendation based on search keywords. In *2012 Ninth Web Information Systems and Applications Conference* (pp. 67–70). IEEE.

Zamani, H., Dadashkarimi, J., Shakery, A., & Croft, W. B. (2016). 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.

Zeng, Y. (2018). Danger, trauma, and verification: eyewitnesses and the journalists who view their material. *Media Asia*, *45*.

Zheng, X., & Sun, A. (2019). Collecting event-related tweets from twitter stream. *Journal of the Association for Information Science and Technology*, *70*, 176–186.

Zhou, X., Cao, J., Jin, Z., Xie, F., Su, Y., Chu, D., Cao, X., & Zhang, J. (2015). Real-time news cer tification system on sina weibo. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 983–988). ACM.

Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, *53*, 1–40.

Zhou, X., Zafarani, R., Shu, K., & Liu, H. (2019a). 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.

Zhou, Z., Guan, H., Bhat, M. M., & Hsu, J. (2019b). Fake news detection via nlp is vulnerable to adversarial attacks. *arXiv preprint arXiv:1901.09657*, .