Expert Systems With Applications 201 (2022) 117027

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Iterative query selection for opaque search engines with pseudo relevance feedback

Maor Reuben [a,b,*], Aviad Elyashar [a,c], Rami Puzis [a,b]

[a] Telekom Innovation Laboratories, Beer-Sheva, Israel
[b] Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Israel
[c] Department of Computer Science, Sami Shamoon College of Engineering, Beer-Sheva, Israel

ARTICLE INFO

Keywords:
Query selection
Opaque search engine
Pseudo relevance feedback
Fake news

ABSTRACT

Retrieving information from an online search engine is the first and most important step in many data mining tasks, such as fake news detection. Most of the search engines currently available on the web, including all social media platforms, are black-boxes (i.e., opaque) supporting short keyword queries. In these settings, it is challenging to retrieve all posts and comments discussing a particular news item automatically and on a large scale.

In this paper, we propose a method for generating short keyword queries given a prototype document. The proposed *iterative query selection* (IQS) algorithm interacts with the opaque search engine to iteratively improve the query, by maximizing the number of relevant results retrieved. Our evaluation of IQS was performed on the Twitter TREC Microblog 2012 and TREC-COVID 2019 datasets and demonstrated the algorithm's superior performance compared to state-of-the-art. In addition, we implemented IQS algorithm to automatically collect a large-scale dataset for fake news detection task of about 70K true and fake news items. The dataset, which we have made publicly available to the research community, includes over 22M accounts and 61M tweets. We demonstrate the usefulness of the dataset for fake news detection task achieving state-of-the-art performance.

1. Introduction

Every day, millions of people search for information online (Manning et al., 2008). For example, market researchers search for products related to their products or business (Yao et al., 2012). Researchers search the academic literature for studies related to their current work (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 other similar cases when additional information related to a specific document is required necessitating online search. In this paper, we refer to such specific documents as *prototypes*, whereas the results that comes up from a search engine as *retrieved documents*.

There are multiple methods used to retrieve a set of documents that are similar to a given prototype from a corpus. Most 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 treat each document as a bag-of-words (Alvarez & Bast, 2017). Advanced methods, such as doc2vec (Le & Mikolov, 2014), skip-thoughts (Kiros et al., 2015), and sent2vec (Pagliardini et al., 2018), use artificial 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 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 engines like Google, or platform-specific search engines, such as Twitter) are opaque, providing very little information about their repositories and algorithms (Jurgen Koenemann & Belkin, 1996). Other than Google's image search, current search engines do not provide document-based search services. Therefore, users usually resort to using short keyword queries which are the mainstay of anyone using search engines today (Chirita et al., 2007).

* Corresponding author at: Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Israel.
E-mail addresses: maorreu@post.bgu.ac.il (M. Reuben), aviadel2@sce.ac.il (A. Elyashar), puzis@bgu.ac.il (R. Puzis).

[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 not use contextual information and retrieve all results exactly matching the specified keywords (Twitter, 0000). Personalization and context aware information retrieval are out of the scope of this paper.

https://doi.org/10.1016/j.eswa.2022.117027
Received 11 March 2021; Received in revised form 12 December 2021; Accepted 27 March 2022
Available online 9 April 2022

Short keyword queries can be ambiguous, failing to 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 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. While it is possible to manually formulate a search query from the document's content (Zhou et al., 2015), 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 for formulating a query miss many relevant results. We further discuss the pros and cons of existing query selection methods in Section 2.

In this paper, we propose a novel iterative algorithm that selects queries that maximize the number of relevant documents retrieved. This approach which require just limited interaction with an opaque search engine consists of two components: the *iterative query selection (IQS)* algorithm and the *word mover's distance (WMD)* measure. The IQS is a hill climbing algorithm that iteratively optimizes short keyword queries given a prototype document, and the WMD provides pseudo relevance feedback by ranking the results of incumbent queries generated by the IQS algorithm according to their relevance to the prototype document. In the absence of a prototype document, incumbent results are compared to a set of relevant results *based on* user relevance feedback. We evaluated the proposed methods on the 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, which can be used *to train* fake news classifiers (see Section 4.5).

The contributions of this paper are:

- we present the *iterative query selection* (IQS) algorithm, an automated mechanism for optimizing short keyword queries (see Section 3.2). The IQS algorithm outperforms existing opaque relevance feedback search on the Twitter TREC Microblog 2012 and on TREC-COVID 2019 datasets (see Section 4.3).
- we compiled a large-scale *Fake News* dataset[2] consisting of 70k news items discussed by 20M Twitter users in 61M tweets (see Section 4.5.1).
- we demonstrated the dataset's quality by applying fake news detection algorithms on the collected data achieving an AUC[3] of 0.92 and accuracy of 0.86 (see Section 4.5.4).

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 describe 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, *and* discuss the results obtained. we conclude the paper in Section 5 with our plans for future work.

2. Related work

This paper presents a novel iterative algorithm that selects queries that maximize the number of relevant documents retrieved. In the subsections that follow, we provide an overview of existing query selection methods. 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 the extraction of 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 studies suggested analyzing the underlying corpus of the given search engine and used this valuable information to expand the queries. Roy et al. (2016) selected the most similar terms for a given query for expansion based on word embedding trained on the corpus, with the aim of choosing the terms that yielded the highest probability of being related to the current query. Kuzi et al. (2016) also used word embedding trained on the corpus to select query expansion terms and suggested centroid- and fusion-based terms and scoring methods used to select them. Xu et al. (2018) selected candidate terms for query expansion based on context features, such as TF-IDF and the co-occurrence of the query terms. Then, the term-ranking models learned were used to rank the candidate terms. Pang and Du (2019) utilized the click-through data of old queries for query reformulation. First, they constructed a click-through network that consists of queries as nodes and edges represent pairs of queries share co-click on the same documents. Then they calculated the conditional probability of each term from the neighboring queries to be in the input query. Finally, they used the top terms to expand short queries and the tail terms to reduce long queries.

All of these approaches require knowledge about the underlying corpus of the search engine, and therefore they are not suitable for opaque search engines. In addition, these approaches require an initial query and 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, an external query selection method is required. 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 a retrieval system presented by Makki et al. (2018). This system is an 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. The ambiguous retrieval strategy sends tweets that have a similar probability of relating to more than one debate for labeling. In the near-duplicate strategy, tweets that have similar text are given the same label. The leveraging hashtags strategy filters tweets containing hashtags that appear in multiple debates. In the leveraging replies strategy, tweets whose 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 considered the query and the retrieved pseudo-relevant documents as users, and the terms as items. Then they used non-negative matrix factorization to recommend terms for the given query. Another approach for query reformulation was introduced by Al-Khateeb et al. (2017), in which 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 and 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

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

[3] Area under the receiver operating characteristic curve.

method that selects effective expansion terms using a random forest classifier 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.

All of these approaches, except for ATR-Vis, require an initial query. The drawback of ATR-Vis is that it requires users to label the retrieved documents (relevance feedback). In contrast, our proposed method uses a pseudo-relevance feedback process that does not require user interaction.

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.

Document similarity measures can be divided into two groups: the supervised measures and the unsupervised measures. Supervised measures require training to provide a similarity score for a pair of documents. Kenter and 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, unlabeled pairs of short texts. The 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's 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.

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. In order to tackle query ambiguity, search result diversification approaches have recently been proposed; these approaches produce rankings to address the multiple possible information needs underlying a query (Drosou & Pitoura, 2010). In most cases, the diversification of retrieved documents implies a trade-off between having more relevant retrieved documents 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 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 retrieved documents. 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 growing 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 classification is very challenging, researchers have suggested various approaches to confronting this issue. Many studies presented approaches based on natural language processing (Zhou, Guan et al., 2019; Zhou & Zafarani, 2020), while others investigated the diffusion of news (Vosoughi et al., 2018; Zhou & Zafarani, 2020). 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, an approach in which tweets are collected based on sources may be incomplete, since there may be many posts are associated with the given claim, but do not contain a link to the claim's source. Moreover, URL shortening, quotes, and cross-referencing are common in the press, as well as among bloggers, leading 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, the use of the source's headlines does not always adequately reflect the claim's content (e.g., in the case of clickbait) which can lead to irrelevant retrieved documents. These drawbacks limit the ability to compile a high-quality dataset that contains enough relevant data for accurate classification.

The second method that people use to collect relevant posts associated with a given claim 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

[4] https://www.snopes.com/.
[5] https://www.politifact.com/.
[6] https://www.factcheck.org/.
[7] https://www.weibo.com/.

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 classification 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.

Due to the limitations of both methods described above, it is clear that fake news classification 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.

3. Iterative query selection with word mover's distance objective function

In this section, we present our novel iterative algorithm for optimizing short keyword queries, given a prototype document, using 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 retrieved documents to a given prototype document. This measure is calculated by summing the shortest distances between words in the prototype document and words in the retrieved documents. (see Section 3.1). The lower the WMD, the more relevant the retrieved documents are. Then, we describe the *iterative query selection (IQS)* algorithm, which used to find queries that retrieve documents with the lowest *WMD* score (see Section 3.2).

3.1. Word mover's distance

In this subsection, we discuss the WMD measure 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 in the retrieved documents and the prototype document. The intuition is that documents that are close in terms of their semantic space are likely to 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 calculated using a pre-trained word embedding method such as GloVe (Pennington et al., 2014), Word2Vec (Mikolov et al., 2013), or FastText (Bojanowski et al., 2017). Any word embedding method in which words with a similar meaning are embedded close to each other can be used.

Let $dist(w_i, w_j)$ denote the cosine distance between the vector representations of w_i and w_j ($w_i, w_j \in \mathbb{R}^n$). Cosine distance is commonly defined as $1 - \frac{w_i \cdot w_j}{\|w_i\|\|w_j\|}$, where $w_i \cdot w_j$ is the dot-product of the two word vectors and $\|w_i\|$ and $\|w_j\|$ are their Euclidean norms. Thus, the cosine distance ranges from zero to two. 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. 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 of the words in W_d (see Eq. (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 of the words in W_d reflects the semantic similarity of the word w_{r_i} to the prototype document. The smaller the distance the greater the semantic similarity.

Given a result document r, let the WMD of r with respect to the prototype document d be the average distance of all words $w_{r_i} \in W_r$ to document d (see Eq. (2)) where $|W_r|$ represents the number of words in W_r (not including stop words).

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

It is important to note that the removal of stop words does not impact the rationality of the proposed method. However, the fact that there are mutual stop words in the retrieved document and the prototype document does not indicate that the documents are similar semantically and are 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 MMD measure for estimate the similarity of the multiple retrieved documents to a given prototype document. Let R be a set of documents retrieved from a search engine. We define the MMD as the mean WMD of all retrieved documents $r \in R$ with respect to the prototype d (see Eq. (3)):

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

The MMD defined above is designed to measure only one aspect of query performance: the relevance of the retrieved documents. Other important aspects, for example, the number of retrieved documents, 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, we recommended word embedding models trained on large non-domain specific datasets.

3.2. Iterative query selection

The proposed IQS algorithm is based on a local search algorithm that selects the queries that maximize the relevance of the retrieved documents from an opaque search engine. We use the hill climbing algorithm, since querying the search engine is resource-intensive, and we need to find the local optimum with just 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 words from which possible queries $q \in V_d$ are selected. V_d may be equal to W_d or it can be 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*); and (2) adding the k closest words in the embedding space for each candidate word in W_d (later referred to as *KNN*).

The IQS algorithm 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 respectively the minimal and the maximal number of words in a query. In each iteration, we randomly modify the query using one of the following three actions: (1) AddWord(q, V_d), which randomly adds a word from V_d that is not yet in query q to q; (2) RemoveWord(q, V_d), which removes a random word from query q, thereby decreasing its size; and (3) SwapWords(q, V_d), which exchanges a random word in query q with a random word in V_d that is not already in q. The possible action at each iteration are based on the query size constrains.

After modifying query q using one of the three actions, we evaluate the MMD of the retrieved documents R_q from the search engine *se*.

Due to computational and network performance considerations, it is important to limit the number of documents retrieved from se in each iteration of the algorithm. Usually, this limit $rlimit$ is defined by the search engine interface and is set to the number of retrieved documents on a single page. The larger $rlimit$ is, the more accurate the $MMD(R_q, d)$ since it will be calculated on more retrieved documents. However, $rlimit$ is also the primary factor (linearly) affecting the run time of an iteration.

The IQS algorithm is implemented as described in Algorithm 1. As input, it receives 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 retrieved documents $rlimit$ in each iteration. To avoid local minimums, we also added the parameter $runs$, which controls the number of times we execute the IQS algorithm on a given prototype document. Since the algorithm is greedy, we only store the queries that decrease the MMD score. If the query returns no results, we consider the query as irrelevant by setting its $MMD(R_q, d)$ score to the maximal score of two. The IQS algorithm returns an ordered set of queries.

Some search engines allow words from the query to be missing in the retrieved documents, while others like Boolean search engine only retrieve documents containing all of the keywords in the query. Twitter is an example of the latter. In the case of a Boolean search engine, it is important to try multiple slightly modified queries in order to retrieve as many relevant results as possible. This is the main reason why the IQS algorithm returns a set of queries and not just the single best query.

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)

4. Experiments

In this section, we describe the series of experiments we performed to evaluate our iterative query selection (IQS) algorithm. Since the algorithm requires a loss function that evaluates each query generated, we first assess the ability of the WMD measure to differentiate between relevant and irrelevant retrieved documents (see Section 4.2). Then, we examine the WMD's correlation to how informative is the prototype document. After evaluating the WMD, we examine the performance of the IQS algorithm as an active retrieval method for an opaque search engine, using the MMD (see Section 4.3).

4.1. Datasets

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

4.2. WMD evaluation

The WMD's purpose in the IQS algorithm is to rank documents based on their relevance to a prototype document. However, the Twitter TREC Microblog 2012 and TREC-COVID 2019 datasets include topic definitions that cannot be used as prototype documents (initial query), due to their rather short length. Therefore, we iteratively construct prototype documents for each topic using a relevance feedback process and evaluate how the ability of WMD to rank documents as relevant. In the rest of the paper, we will refer to document ranking methods, such as WMD, as relevance measures.

4.2.1. Experimental setup

The prototype document generated should reflect the topic being searched. We use the following process to iteratively build a suitable prototype document for each topic and improve the documents retrieved using the WMD. First, we use the initial query for each topic as the prototype document, and we calculate the WMD between the prototype document and each tweet in the dataset. Second, we retrieve the top k documents and request relevance feedback from a user (or an oracle if the ground truth is provided for evaluation purposes). Next, we expand the prototype document using the content of the relevant documents retrieved and perform the second step again.

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

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

[8] https://www.kaggle.com/c/trec-covid-information-retrieval/overview.
[9] https://www.semanticscholar.org/cord19.

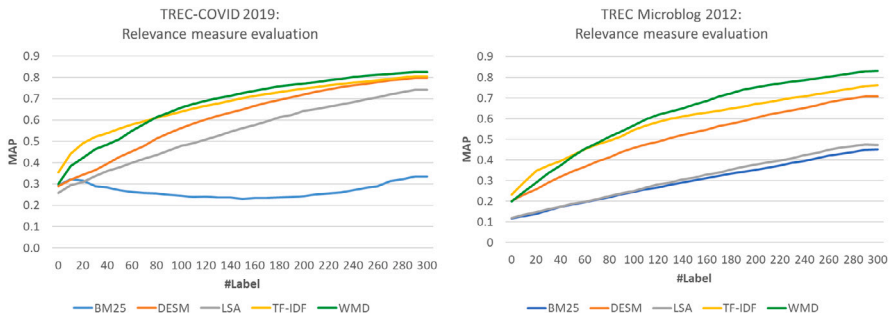

Fig. 1. Evaluation of the relevance measures on the Twitter TREC Microblog 2012 and TREC-COVID 2019 datasets relative to the number of labels received from the relevance feedback process.

Table 1
Comparison of the performance of the relevance measures using relevance feedback on the TREC microblog 2012 and TREC-COVID 2019 datasets.

Method	TREC microblog 2012		TREC-COVID 2019	
	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**

4.2.2. Results & discussion

The mean average precision (MAP) and R-precision results are presented in Table 1. As can be seen, the WMD outperforms the other methods evaluated on both datasets in terms of both the MAP and R-precision. These results highlight the WMD's effectiveness as a means of distinguishing between relevant and irrelevant documents. The results also show that pre-trained word vectors are very useful for detecting similar words in two documents.

We also examine the effect of the prototype document's informativeness (based on the number of labels) on the MAP score. This aspect is important, because it indicates whether the relevance measures accurately estimates the retrieved documents' relevance according to the prototype or not. We compared the relevance measures on both TREC Microblog 2012 and TREC-COVID 2019 datasets and presented the results in Fig. 1. The trends in the results show how well the WMD utilizes the information found in the prototype document to rank the retrieved documents, as it outperforms the other methods on this task. This finding indicates that the WMD is the best relevance measure candidate for our query selection method.

4.3. Iterative query selection with relevance feedback

In this experiment, we evaluate the full IQS pipeline based on a process that mimics interaction with Twitter's search engine. Twitter uses a Boolean retrieval model, which means that the retrieved documents must contain all of the words in the query. We assume an opaque search engine se and access the corpus through the Boolean search process, as in Twitter.

4.3.1. Experimental setup

Similar to the previous experiment, we iteratively construct a prototype document for each topic using relevance feedback. In this case, we use the topic definition as the prototype document. We then run the first iteration of the IQS algorithm calculating the MMD score of the prototype document and the retrieved documents. Before proceeding to the next iteration of the IQS algorithm, we sort the documents retrieved in ascending order, according to their MMD score. Then, we take the top $k = 10$ retrieved documents and ask a user (or oracle) to label them. Next, we expand the prototype document using the content of the relevant retrieved documents identified by the user (or oracle) and proceed to the next iteration of the IQS algorithm. The stopping condition is the same as in the previous experiment, reaching $n = 300$. We set the minimal and maximal number of words in a query to be between one and six ($minq = 1, maxq = 6$). This parameter directly influences the number of documents retrieved. When a query contains a single word, we can expect that just some of the retrieved documents will be directly relevant to the prototype document. For example, assume that the prototype document is "Crude oil production in the US" and the query only contains the word "oil". In this case, the Twitter search engine will likely retrieve many tweets that include the word although they are not directly related to the US oil industry (for example, it may also retrieve tweets that focus on oil painting and oil production in Russia). In the same manner, a large number of words in a query decreases the number of documents retrieved, although most of the tweets will be relevant to the given prototype document. Lastly, we set the number of retrieved documents to 20 ($rlimit = 20$) to simulate the behavior of a standard search engine on the web.

In order to select the best hyper-parameters for IQS algorithm, we examined several ranges: itr ranging from 10 to 45, $runs$ ranging from one to three, and $numQueries$ ranging from five to 50. In 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 requests to the search engine ($runs * itr$) to a maximum of 45, due to time constraints. Each request to the Twitter search engine takes approximately 1.4 s. Therefore, the total run time for each prototype document is about a minute. Eventually, the best parameters found for the IQS algorithm were $itr = 15$, $runs = 3$, $minSize = 1$, $maxSize = 6$, and $numQueries = 40$.

4.3.2. Results & discussion

We compared the performance of the IQS algorithm to the ReQ-ReC implementation available on GitHub,[10] using the same settings: the top 10 documents are labeled by the user ($k = 10$), and the algorithm stops after are obtained 300 labels for each topic ($n = 300$). In addition, we compare our method's performance to that of the ALMIK method proposed by Zheng and Sun (2019) on the TREC Microblog 2012 and TREC-COVID 2019 datasets. We implemented the ALMIK method based on the description provided in the original paper. Again, we limit the number of label requests from the user to 300. In order to achieve the best results, we also performed hyper-parameter tuning; the best results were achieved when there were three rounds of the active learning phases. Between the phases, we included a keyword expansion phase and used the new retrieved documents in the next round. In each active learning phase, we performed 10 iterations of 10 label requests for the most uncertain tweets from the user (for a total of 100 labels in each active learning phase).

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

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

Method	TREC microblog 2012		TREC-COVID 2019	
	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 reports the performance of the IQS algorithm, ReQ-ReC, and ALMIK on both datasets in terms of the MAP and R-Precision. As seen in the table, the proposed IQS algorithm outperformed the ReQ-ReC and ALMIK methods in terms of the MAP and R-Precision on both datasets. The results showcase the IQS algorithm's ability to retrieve more relevant documents from a Boolean opaque search engine, given a short initial query, using relevance feedback.

4.4. Iterative query selection with keyword expansion

In the above experiments, we used the *vanilla* IQS algorithm without the addition of any query expansion methods. In this experiment, in addition to the vanilla configuration, we also used two keyword expansion methods associated with the IQS algorithm. In the first method, each candidate word is expended using its top five synonyms from WordNet (IQS+Syn). In the second method, each candidate word is expanded using its five nearest neighbors (i.e., words) based on *FastText* word embedding (IQS+KNN). We compare the configurations in terms on number of iterations and number of labels from the user. It is important to examine how many iterations it takes for each configuration to converge since we want to limit the number of requests to the search engine, and we also want to examine how the number of labels affects the convergence.

4.4.1. Experimental setup

To evaluate the performance (i.e., MAP) of the three configurations (IQS, IQS+Syn, and IQS+KNN) as a function of the number of iterations, we executed each configuration with a maximum of 85 iterations while assessing the performance after every five iterations. We run each of the configurations with a relevance feedback process of 50 labels at each feedback step. Since there are cases in which the query with the lowest score retrieves irrelevant documents, we used the top five queries to decrease the variation in performance. We done this process three times for each configuration and report the average performance. The results are presented in Fig. 2 shows the MAP score on each configuration after 0, 50, 100, 150, 200, and 250 relevance feedback labels.

4.4.2. Results & discussion

As can be seen in Fig. 2, the performance of the vanilla IQS configuration was found to be superior until a turning point was reached, at which point the IQS+Syn and IQS+KNN outperformed the vanilla IQS configuration. We also can see that the vanilla IQS converged faster than the other configurations. This is expected, since the vanilla IQS has a smaller search space than the IQS+KNN and IQS+Syn configurations, i.e., the algorithm requires fewer search iterations to converge. We can also see that the number of relevance feedback labels affects the performance of the configurations. The more relevance feedback labels the sooner the turning point arrives. The reason for this is that the ratio between the number of keywords in the original prototype and the number of keywords in the expanded prototype decreases when the prototype document is more informative. A low ratio means that there is a higher likelihood of using more original keywords and thus performing similarly to the vanilla configuration at the start. Since we want to keep the number of interactions with the search engine low, the vanilla IQS configuration is the best option.

4.5. Application for fake news detection with pseudo relevance feedback

In many cases, it is not practical or scalable to ask for continuous user feedback. Therefore, in this experiment, we are in a mode of pseudo-relevance feedback, meaning that we apply the IQS algorithm with the MMD to provide pseudo-relevance feedback. To reduce the number of interactions with the Twitter search engine, we use the IQS vanilla configuration.

In this subsection, we demonstrate the use of the IQS algorithm as a step in the fake news detection pipeline. We use the proposed IQS and MMD method for the automated collection of relevant tweets associated with labeled news items (the ground truth). Later, using these relevant tweets, we demonstrate fake news detection using supervised machine learning classifiers. First, we describe the data collection process and provide some background on fake news detection on online social media (OSM). Then, we present the dataset collected using the IQS algorithm for fake news detection task. Finally, we train a classifier on the collected data and present its performance.

4.5.1. Data collection with the IQS algorithm

In this subsection, we describe the compilation of a large dataset for the task of fake news detection on OSM using the IQS algorithm. 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 labels: true, mostly true, false, mostly false and pants on fire. Similar to Rasool et al. (2019), we converted the classification problem into a binary task by categorizing the news items that their label are true and mostly true as true and news items labeled false, mostly false and pants on fire as false. For Gossip Cop, we categorized news items with scores between zero and three as false and news items with scores between seven and 10, as true. Since the majority of news items on fact-checking websites are false, we added news items from 10 well-known news sources (*The Times 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 have exploited reliable news sources as a proxy for true news, e.g., items (Monti et al., 2019).

We collected a total of 70,018 news items (16,212 false and 53,806 true). For each news item, we set the IQS algorithm to run on the Twitter API three times, with the following parameters: five final queries returned ($numQueries = 5$), a maximal number of 15 iterations ($itr = 15$), the number of words in a query is between three to six ($minq = 3, maxq = 6$), and the number of retrieved documents equals 20 ($rlimit = 20$). Then, after obtaining the top five final queries from all three runs, we used them in order to retrieve the most relevant tweets for each news item, limiting the number of tweets returned for each query to 500. To make the fake news detection more realistic, we only collected the tweets that were posted before the fact-checker assigned a label to the news item. Utilizing this approach, we compiled a large fake news dataset[12] containing around 70,000 news items and about 61 million corresponding posts, the distribution of which is shown in Table 3.

4.5.2. Fake news background

The task of fake news classification has been extensively studied in recent years. Along with the growth of online news, many non-traditional news sources, such as blogs, have evolved to response 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

[11] https://www.gossipcop.com/.
[12] https://drive.google.com/drive/folders/1nOGYjGoZHxFwaPm0xci-T7V0a90YW448?usp=sharing.

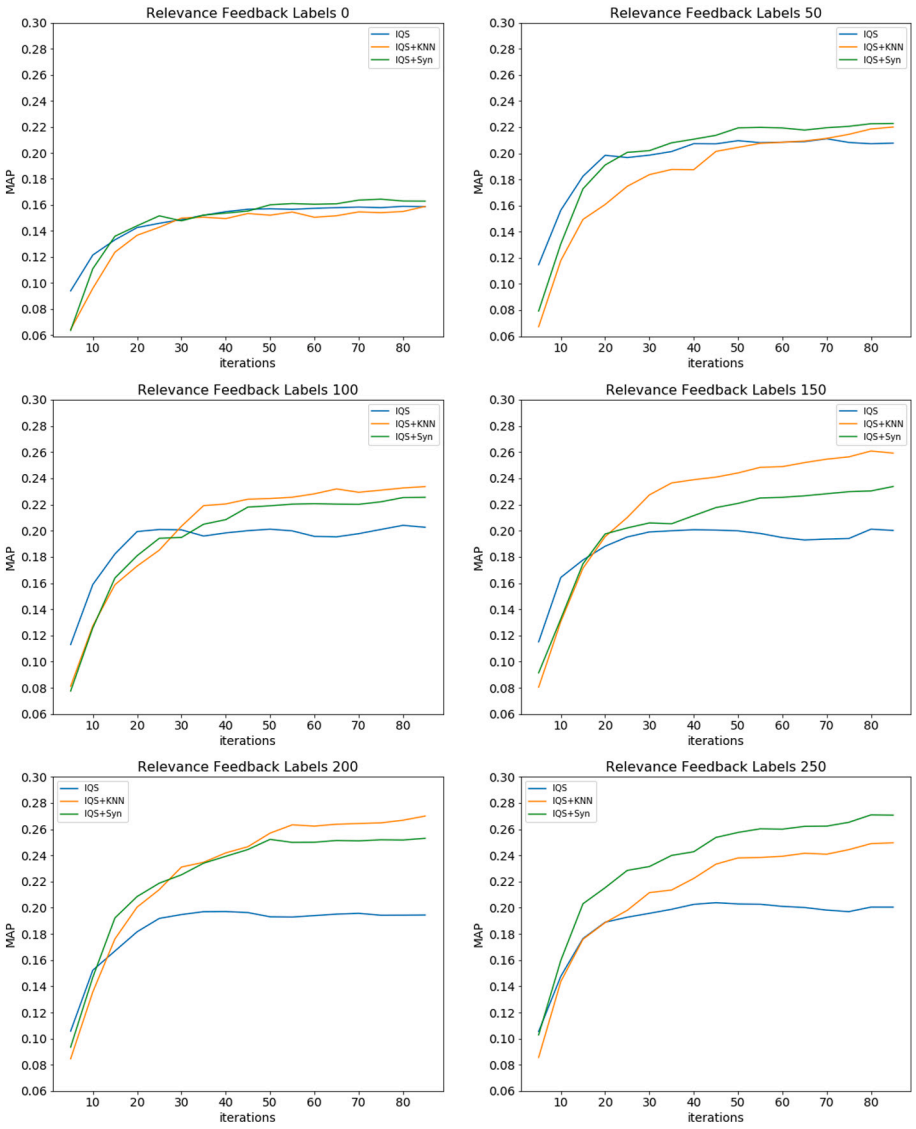

Fig. 2. Evaluation of the three IQS configurations' performance on the Twitter TREC Microblog 2012 dataset. Each graph presents the MAP score after 0, 50, 100, 150, 200, and 250 relevance feedback labels. The MAP score is calculated based on the top five selected queries returned from the IQS configuration.

Table 3
Fake news dataset statistics.

Domain	#News items	#True	#False	#Authors	#Tweets
politifact.com	12,952	5288	7664	4,274,688	12,525,467
snopes.com	4682	845	3837	1,721,162	3,617,550
gossipcop.com	4764	53	4711	863,906	2,302,245
Times of Israel	21,989	21,989	0	5,401,027	20,963,687
Jerusalem Post	3109	3109	0	1,469,181	3,386,702
New York Times	3453	3453	0	1,641,931	3,247,247
ABC News	3685	3685	0	1,042,084	2,410,832
CNN News	3496	3496	0	1,381,585	2,988,373
Fox News	3620	3620	0	1,115,282	2,921,356
BBC News	3687	3687	0	1,181,490	2,469,421
MSNBC	2410	2410	0	731,839	1,982,974
Politico	1640	1640	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**

unverified news and the absence of quality control procedures to prevent potential deception, has contributed to an ever-increasing problem of fake news dissemination (Conroy et al., 2015). The spread of misinformation, propaganda, and fabricated news has potentially harmful effects and has had significant impact on real-world events (Allcott & Gentzkow, 2017), such as the Brexit referendum and the 2016 US election, and weakened public trust in democratic governments (Zhou, Zafarani et al., 2019). The global economy, like local economies, is not immune to the impact of fake news; this was demonstrated when a false claim regarding an injury to President Obama caused the stock markets to plunge (dropping by 130 billion dollars) (Rapoza, 2017). In recent years, due to the threats to democracy, journalistic integrity, and the economy, researchers have been motivated to develop solutions for this serious problem (Zhou, Zafarani et al., 2019), examining and proposing approaches for fake news detection based on natural language processing (Zhou, Guan et al., 2019), investigating the diffusion of news (Vosoughi et al., 2018) and more.

4.5.3. Fake news classification

In the data collection section (see Section 4.5.1), we describe how we compiled a large dataset for fake news classification and in this section we describe the classification process. To classify the news items, we extracted author- and post-based features. For author-based features, we applied aggregation functions on various aspects of author demographics, such as the registration age, number of followers, number of followees, number of published tweets, etc. Post-based features

Table 4
Features listed according to Gini importance.

No.	Feature name	Gini importance
1	Number of verified authors	0.162
2	Maximum number of posts published by users	0.049
3	Glove_wikipedia_model_300d max dimension 295	0.046
4	Glove_wikipedia_model_300d min dimension 291	0.037
5	Maximum favorite count	0.036
6	Maximum follower count	0.033

include the aggregations of posts' metadata, such as retweet count, text length, and the time interval between the oldest and newest post. We removed stop words from all of the features extracted from the post. With regard to the posts' data, we extracted the following features: sentiment, temporal (post diffusion patterns), LDA[13] (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 the mean, median, max, min, standard deviation, kurtosis, and skewness functions.

4.5.4. Results & discussion

For 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 of the folds. We determined that the best performing classifier on the test set was the random forest with 100 estimators and a maximum depth of 10. This classifier with 100 features obtained an AUC of 0.92 and accuracy 0.86. These results show the benefit of using data collected based on the IQS algorithm for fake news detection on OSM.

We also analyzed the most influential features of the best-performing 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 verified authors with respect to fake and true news items, we can see that for true news items, the number of verified authors is three times higher than for false news items. These difference were found to be statically significant (a p-value of 0.0). Based on this result, we conclude that verified authors are important actors in fake news detection. The greater their participation in online discourse, the more reliable the content of the online discussion. In terms of the Gini importance, the second, fifth, and sixth most influential features were aggregations on the news item's authors. This strengthens the conclusion drawn by Castillo et al. (2011) that author-based features are very relevant for fake news detection on OSM. In addition, the third and fourth most influential features were aggregations on the word embedding of the news item's posts. These findings indicate that the vector representations of the words consist of the online discussions, can imply the truthfulness of given news items.

These results demonstrate that our proposed IQS algorithm can be utilized for solving real-world problems (e.g., the classification of fake news). In addition, the machine learning classifiers trained on the large dataset collected using the IQS algorithm obtained impressive results. This strengthens our conclusion that our method can be very useful for classifying fake news.

5. Conclusion & future work

In this study, we proposed 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 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).

To evaluate the proposed method, we used the Twitter search engine to retrieve tweets associated with the given prototype document. This service collects tweets published by accounts that agreed to share their information publicly. Our evaluation of IQS algorithm on the *Twitter TREC Microblog 2012* and *TREC-COVID 2019* datasets and comparison to two state-of-the-art methods demonstrated the proposed algorithm's superiority on both datasets. Next, we applied IQS algorithm to produce a large fake news dataset which we later successfully used for the task of fake news detection. Based on our examination of the proposed algorithm and its application on a real-world problem, we draw the following conclusions: First, the WMD score can be used to differentiate relevant and irrelevant documents for a given prototype document (see Section 4.2). This result strengthens the conclusions of Kusner et al. (2015) who found it to be effective for document classification. Second, the IQS algorithm outperformed two state-of-the-art methods: ReQ-ReC (Liu et al., 2014) and ALMIK (Chy et al., 2019) and was found to be effective for the task of retrieval with relevance feedback. Third, the use of the proposed IQS algorithm in an automated fake news detection pipeline is recommended. We used the algorithm to compile a large-scale news dataset consisting of about 70k true and fake news items. The dataset, publicly available for research, includes more than 22M accounts and 61M tweets. 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 the IQS algorithm.

There are a few limitations associated with the proposed IQS algorithm. In cases in which the prototype document is associated with a general topic that includes several sub-topics, for example, *US presidential elections*, the algorithm would probably narrow the focus and retrieve documents related to a specific sub-topic (e.g., claims associated with voter fraud during the 2016 election). This would occur, since the algorithm searches for a local optimum, which means it converges to the first one topic it discovers. In addition, in cases in which large documents are retrieved from a search engine, the MMD will not work well due to the high computational time required for the WMD measure.

Considering ethics, collecting public information from OSM has raised ethical concerns in recent years; however, 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.

One possible direction for future work is to demonstrate the proposed approach on other OSM platforms, such as Reddit[14] and Quora.[15] In addition, future work could compare the effectiveness of a fake news detection system using data collected using a source URL versus data collected using our query selection method.

CRediT authorship contribution statement

Maor Reuben: Ideas, Writing – original draft, Writing – review & editing, Methodology. **Aviad Elyashar:** Ideas, Writing – review & editing, Reviewing and editing. **Rami Puzis:** Supervision, Ideas, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[13] Latent dirichlet allocation.

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

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

Acknowledgment

The authors would like to thank Robin Levy-Stevenson for editing this article.

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*(2), 211–236.

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

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).

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.

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*(6), 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*(1), 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*(4), 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*(3), 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.

Jurgen 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.

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).

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*(1), 3.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

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*(11), 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*(3).

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. 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*(1), 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, vol. 2012* (p. 20). Citeseer.

Twitter. (0000). 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*(6380), 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)* (p. 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*(1–2).

Zheng, X., & Sun, A. (2019). Collecting event-related tweets from twitter stream. *Journal of the Association for Information Science and Technology*, *70*(2), 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, Z., Guan, H., Bhat, M. M., & Hsu, J. (2019). Fake news detection via NLP is vulnerable to adversarial attacks. arXiv preprint arXiv:1901.09657.

Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys*, *53*(5), 1–40.

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