Efficient Outlier Detection in Text Corpus Using Rare Frequency and Ranking

WATHSALA ANUPAMA MOHOTTI, Queensland University of Technology, Australia and University of Ruhuna, Matara, Sri Lanka
RICHI NAYAK, Queensland University of Technology, Australia

Outlier detection in text data collections has become significant due to the need of finding anomalies in the myriad of text data sources. High feature dimensionality, together with the larger size of these document collections, presents a need for developing accurate outlier detection methods with high efficiency. Traditional outlier detection methods face several challenges including data sparseness, distance concentration, and the presence of a larger number of sub-groups when dealing with text data. In this article, we propose to address these issues by developing novel concepts such as presenting documents with the rare document frequency, finding ranking-based neighborhood for similarity computation, and identifying sub-dense local neighborhoods in high dimensions. To improve the proposed primary method based on rare document frequency, we present several novel ensemble approaches using the ranking concept to reduce the false identifications while finding the higher number of true outliers. Extensive empirical analysis shows that the proposed method and its ensemble variations improve the quality of outlier detection in document repositories as well as they are found scalable compared to the relevant benchmarking methods.

CCS Concepts: • Theory of computation → Unsupervised learning and clustering; • Applied computing → Document management and text processing;

Additional Key Words and Phrases: Outlier detection, high dimensional data, k-occurrences, ranking function, term-weighting

ACM Reference format:
Wathsala Anupama Mohotti and Richi Nayak. 2020. Efficient Outlier Detection in Text Corpus Using Rare Frequency and Ranking. ACM Trans. Knowl. Discov. Data 14, 6, Article 71 (October 2020), 30 pages. https://doi.org/10.1145/3399712

1 INTRODUCTION
With the advances in data processing technology, digital data have witnessed exponential growth [27]. Outlier detection plays a vital role in identifying anomalies in massive data collections. Some useful examples are credit card fraud detection, criminal activity detection, and abnormal weather prediction [40]. The general idea of outlier detection is to identify patterns that do not conform to general behavior, referred to as anomalies, deviants or abnormalities [1]. There exist different
supervised and unsupervised machine learning methods that have been used to identify exceptional points from different types of data, such as numerical, spatial, and categorical.

Outlier detection in text data is gaining attention due to the generation of a vast amount of text through big data systems. Reports suggest that 95% of the unstructured digital data appears in text form [27]. Digitization of news and contextualisation of user interaction over social blogging result in larger text repositories that exhibit multiple topics. Text outlier detection plays an important role in these repositories to identify anomalous activities that can be malicious and informative. An outlier text document has content that is different from the rest of the documents in the corpus that share a few similarities among them [4].

Figure 1 illustrates this scenario where a repository contains several sub-groups of documents (called as inliers) and some documents that do not belong to any of the subgroups (called as outliers). Identifying this type of outliers is beneficial in many application domains for decision-making such as web, blog, and news article management [31]. A subset of unusual web pages on a website or blog pages on a blog deviating from the common themes in the website/blog, if discovered, will draw useful insight for administrative purposes. Similarly, detecting mis-matched news articles from a collection of news documents may help to flag them as emerging, exceptional or fake news [1]. The unusual events detection from the (short-length) social media data can indicate early warnings [15]. Therefore, accurately identifying outlier documents that are dissimilar to inlier sub-topics within a larger document collection in an efficient manner is important for text outlier detection domain.

The existing applications of identifying text outliers face several challenges. (1) Unavailability or less availability of labeled data is the primary challenge for outlier detection methods that requires researchers to focus on developing unsupervised methods. (2) Text data usually show fewer co-occurrences of terms among documents and form a sparse representation that presents difficulties to traditional document similarity calculation methods to identify the deviations [31]. (3) There is a special category of text data such as social media text in which the number of discriminative terms and common terms shared by related text is very small. Consequently this data representation becomes extremely sparse [44, 56]. Moreover, the number of subgroups or topics on social media is considerably high. It requires text mining methods to identify the outliers that are deviated from all these groups. Text outlier detection methods face additional challenges in handling short-length social media posts. (4) Moreover, the larger sizes of text collections generated by big data systems create a need to explore efficient but accurate outlier detection methods.
Outlier Detection in Text

Studies on general outlier detection commonly use distribution, distance, and density-based unsupervised proximity learning methods to address the unavailability of training data with class labels. Most of these methods suffer from efficiency problems due to the computation required dealing with the high volume in large datasets [48]. The effectiveness of these methods on high-dimensional data is further challenged by the well-known curse of high dimensionality [40] when applied in text data. Specifically, the distance difference between near and far points becomes negligible and the proximity-based methods cannot differentiate them based on similarity [62]. This challenge is further amplified when the data collection exhibits numerous distinct groups within the collection [53] (as in Figure 1). Specifically, identifying dissimilarities to multiple groups with high dimensional text representation is challenging.

Researchers have developed subspace and angle-based methods to address the problem of high dimensionality in the datasets. However, these methods are known to be computationally expensive due to the larger numbers of comparisons required. Moreover, the subspace analysis cannot guarantee that relevant subspaces are aligned with extreme values in full dimensionality [36, 40]. Another set of solutions are proposed based on nearest-neighbors (NN) and “anti-hub” concepts for high-dimensional data such as graphs, genes, and the like [20, 22, 26, 47]. Though the NN calculation is a successful concept to use in identifying text similarity, it is known to present scalability issues for larger datasets [52].

There are only limited studies in outlier detection literature that specifically focus on text-domain and deal with the sparseness of the document feature vector. Recently, the use of sum-of-square differences in matrix factorization is proposed to determine outlier scores in text data [31]. In real-world scenarios such as social media, the number of groups/topics inherent in the data is large (as shown in Figure 1) and creates the need for distinguishing the fine-grained sub-groups while identifying outliers. However, the fine-grained problem dealing with a larger number of inliers (i.e., normal data) and outlier classes presents issues for the aforementioned matrix decomposition as the iterative lower-rank matrix approximation process increases the level of error (as evident in our experiments).

Information Retrieval (IR) systems that effectively represent documents with a weighting scheme and calculate the similarity between documents through the inverted index data structure avoiding the issues with sparse data representation show a promising direction. Inverse Document Frequency (IDF) is a common statistical measure in IR that selects intrinsic dimensionality of a text document by representing how important a term is in a collection [13]. Inspired by this, we propose a novel rare-frequency-based method for high-dimensional document outlier detection. We propose that semantic term clusters can effectively be used to detect deviations or anomalous documents through meaningful term weighting.

Next, we conjecture that the relevant document set retrieved by a search engine, in response to a document (posed as a query), is an alternative solution to generate the neighborhood of the document. We present two novel methods to calculate outlier scores based on this ranking-based neighborhood concept using the scalable search engine technology in comparison to the conventional big data analytics methods that require significant investments. We propose to use the local sub-dense neighborhood concept (Hubs), evident in high-dimensional text data, through ranking.

Finally, we combine the neighborhood-based methods with the primary rare-term weighting-based method to form ensemble approaches and reduce the potential of false outlier detections. Unlike the state-of-the-art methods, we present all of these methods as non-parametric and address the bottleneck of setting the user-defined threshold to assess a document score. Lastly, the article discusses the need for an outlier-focused evaluation mechanism to report false positives (i.e., false outliers) and false negatives (i.e., false inliers) in outlier detection.
In summary, this article presents accurate and efficient outlier detection methods based on the concepts of rare document frequency and ranking brings several novel contributions to the area of document outlier detection, listed as:

— The use of rare frequency in document representation for outlier detection to demarcate the border between common and rare documents. This novel concept contributes to the primary Outlier Detection Based on Inverse Document Frequency (OIDF) algorithm.
— The concept of finding relevant neighbors using a scalable IR system that consumes less computation cost. Two novel algorithms (ORFS and ORNC) are developed to detect the level of deviations between documents.
— A set of ensemble approaches (ORFS (I), ORFS (S), and ORNC (S)) focusing on improving accuracy (i.e., reduced false outliers) with efficiency. Their approaches do not depend on a user-defined parameter as an outlier threshold.
— Envisaging the requirement of meaningful evaluation measures, namely, $OPE$ and $IPE$, to highlight false detection.

The rest of this article is organized as follows. Section 2 provides motivation and related works related to outlier detection, term weighting, and IR ranking concepts. The proposed approaches for text outlier detection based on rare term weighting and ranking are detailed in Section 3. A comprehensive empirical study with benchmarking on several datasets covering various length text data is provided in Section 4, with a summary that provides useful insight on all approaches. Finally, concluding remarks are summarized in Section 5.

2 RELATED WORK

In the current era, most human interactions appear and are collected in the form of free text such as emails, wikis, blogs, and social media feeds. Outlier detection is useful for finding interesting and suspicious text within a collection representing multiple themes.

A text collection usually contains a high-dimensional set of terms that result in a sparse representation [3]. Different text representation schemes based on term frequency (TF) have been used with the Vector Space Model (VSM) [25]. There are different term weighting schemes used in IR to give an importance level to terms such as TF, IDF, TF*IDF, and BM25 [50] in a data model. The IDF scheme favors rare terms in the collection [13]. Several prior works in the field of outlier detection use Hawkins’s definition to set an outlier, stating that "An outlier is an observation which deviated so much from the other observations to arouse suspicions that it was generated by a different mechanism" [23]. These deviations can be identified by using rare terms in the calculation. In this article, we conjecture that the use of IDF, which values the importance of rare words, in presenting a dataset will highlight the outlier documents.

Outlier detection broadly follows two approaches. (1) Supervised learning when training data with labels of normal and abnormal data is provided [34]; and (2) unsupervised learning when labeled data is not available, which is common in real-world scenarios [22]. Neural network-based methods that used deep feature extraction [14], and Generative Adversarial Network (GAN)-based active learning methods in outlier detection [42] are recent supervised and semi-supervised methods that predict outliers based on training data directly or indirectly. The performance of these methods is fully or partially affected by the supervision given by the labelled data.

Unsupervised learning methods follow proximity approaches such as distance-based, density-based, distribution-based, and cluster-based [22]. The majority of these methods deals with a few-dimensional numerical data where overlapping in distance or density distribution separates outliers. Distribution-based methods use different statistical fundamentals to determine the anomalies that occurred outside of the normal model [8, 28]. These methods depend on the assumption about
data representation and measures used and can be affected by over fittings in normal data. Poor scalability of a proximity approach for the high-dimensional data further makes it less effective for text outlier detection.

Distance and density-based approaches have been extensively used in outlier detection due to their simplicity in implementation [40]. Conventional distance-based methods identify outliers that highly deviate from the remaining data in the collection using distance differences [33]. Alternatively, neighborhood information is used for outlier detection [49, 57]. NN has been used as an effective method to measure the distance differences. The differences between each point and $k$-NN are considered and the top $n$ farthest points are labeled as outliers [49]. Text data presents challenges to this approach where distance differences become negligible due to sparseness in high dimensions (known as distance concentration) [40]. Document collections are usually large and contain multiple groups. This creates a scalability issue for nearest-neighbor, calculation-based approaches [53].

As a remedy to the distance concentration problem, similarity calculation based on the angle between vectors is proposed to determine the deviation [37]. This approach can be adapted to the text domain as cosine similarity can be used to measure angle differences in text feature vectors [55]. However, the number of pairwise comparisons needed for larger datasets increases the computational complexity and makes this approach infeasible to apply to large-scale data.

In contrast to the distance-based methods that identify far-away points globally, density-based methods identify less dense points locally as outliers. These methods derive a density distribution of data and identify nearest neighbors by handling varying density patches. The relative density of a point is compared to neighbors and a Local Outlier Factor (LOF) is defined to determine the degree of outlier [11]. A point gets a higher LOF value if the ratio between density around $k$ nearest-neighbors of that point and local neighborhood of that point is high. It is then labeled as an outlier candidate [37]. Density-based methods are known to face difficulties to deal with higher dimensions, inherent to text data due to distance concentration.

Density-based clustering methods such as DBSCAN can naturally detect outliers in the dataset by considering them as points in sparse regions [5, 19]. Several clusters-based outlier detection methods have been proposed focusing the tightness of the clusters [16, 24]. Although these methods are capable of detecting outlier clusters, they highly depend on threshold parameters [16]. Moreover, these methods cannot be directly adapted to text data, as the text data exists in patches and it becomes highly difficult to detect outliers.

In high-dimensional data, identifying the outliers that are deviated from the rest of the collection is difficult with distance and density methods due to less effectiveness of similarity calculation between points with the curse of dimensionality [2]. Different multi-dimensionality scaling techniques have been used to deal with this issue [10] and identify the outliers in reduced dimensions. However, the loss of information in the higher-to-lower order approximation is inevitable. Alternatively, subspaces-based outlier detection methods play a vital role to manage high dimensionality. They combine local data pattern analysis with subspace analysis. However, the process of finding a subset of dimensions, with rarely existing patterns, using brute-force searching mechanisms incurs high computational cost [2]. Furthermore, these approaches rely on selected subspaces to identify outliers in the entire data [36]. The deviations in the embedded subspaces may not determine outliers in full dimensional space globally.

Text data has been shown to experience the Hub phenomenon that is evident in high dimensions, i.e., “the number of times some points appear among $k$-NN of other points is highly skewed” [47]. These local NNs, which form sub-dense regions (i.e., Hubs), are used to address sparseness-related problems in high dimensionality effectively [20, 45, 53]. This concept has been used inversely in outlier detection. In a graph-based method where each data point represents a
Table 1. Summary of the Major Outlier Detection Methods Used in High Dimensional Data

| Category          | Methods                                      | Applied Domain   |
|-------------------|----------------------------------------------|------------------|
| Ranking-based     | Neighborhood-based outlier detection [40]    | Numerical data   |
|                   | k-occurrence-based                           | Numerical data   |
|                   | Anti-hub based                               | Numerical data   |
|                   | Hubness aware                                | Numerical data   |
| Graph-based       | k-NN graph-based                             | Numerical data   |
|                   | Natural-neighbor graph-based method in [26]  | Numerical data   |
| Subspace-based    | Evolutionary algorithms in [2]               | Numerical data   |
|                   | Subspace outlier detection in [36]           | Numerical data   |
| Projection-based  | Random projection                            | Text data        |
|                   | NMF-based outlier detection [31]             | Text data        |
| Angle-based       | Angle variance-based method in [37]          | Numerical data   |

graph node, connections are made considering the reverse $k$-NN [22, 26] and a low in-degree score identifies potential outlier nodes. Similarly, researchers have used the concept of “anti-hubs” as potential outlier candidates [48]. Although these local NNs-based approaches successfully handle high dimensions, the scalability of these methods for larger datasets remains questionable, due to the need for calculating a hub for each data point.

Table 1 summarizes the outlier detection methods that have been applied in numerical and textual datasets. Limited studies exist that focus specifically on the text domain [8]. Given the fact that random projection approximately preserves the distance between points in the lower-dimensional space, the random projection has been applied to text data to identify outliers [6]. Loss of information is inevitable in projection and ultimately reduces accuracy. To improve the accuracy of text outlier detection, the significance of terms should be interpreted related to the structure of the document [1]. Non-negative Matrix Factorization (NMF) has been used to decompose the text collection into a set of semantic term clusters and document clusters considering document structures [31]. The term clusters allowed the method to learn the outlier degree of each document by ranking the sum-of-squares of differences with the original matrix. However, the increased number of groups within the collections makes this learning process impaired. Outlier detection in fine-grained scenarios is not practical with an NMF-based method in terms of both accuracy and scalability.

The IR systems have shown the capacity to manage the text data successfully [3]. Search engines are well-known IR systems capable of efficiently finding relevant documents from a document collection, whereby the document collection is organized in the inverted indexed data structure [59]. They have been known to deal with big data collections [59]. In this article, we propose the novel ensemble approaches based on the rare frequency and ranking concepts in IR, and identify the NNs as well as local sub-dense neighborhoods in text data to determine the deviations. To the best of our knowledge, this is the first extensive outlier detection work in text mining, using the concepts of rare frequency, ranking and ensemble approach.
3 OUTLIER DETECTION IN TEXT DATA: PROPOSED METHODS

3.1 Preliminaries

In this article, the objective of outlier detection is to identify data points distinct from the rest of the collection as outliers, by separating them from the inliers that are cohesive points forming subgroups. Consider a document collection \( D = \{d_1, d_2, \ldots, d_N\} \) where \( d_i \in D \) is represented using a set of distinct terms \( \{t_1, t_2, \ldots, t_v\} \). Let \( D \) contains a set of distinct terms \( \{t_1, t_2, \ldots, t_n\} \), \( n \gg v \) covering all the terms in the collection. Let \( D \) be divided into a set of sub-groups \( C = \{c_1, c_2, \ldots, c_l\} \) where \( l \ll n \) and \( l < N \). Each \( c_q \in C \) contains a set of similar documents that share related terms. We formally define document \( d_i \in D \) as outlier or inlier as follows.

**Definition 3.1 (Outlier).** A document \( d_i \in D \) that shows high deviation, based on term distribution, from all distinct sets of similar documents \( C \) is considered an outlier.

**Definition 3.2 (Inlier).** A document \( d_i \in D \) that shows high similarity, based on terms distributions, with any distinct set of similar documents, \( c_q \in C \) is considered an inlier.

**Example:** Figure 3(a) shows a toy document collection. It contains two (sport) groups of documents considered as inliers, and the document on weather information, \( d_{11} \), is an outlier. The outlier document shows a set of different terms that deviates from the common terms in the collection related to subcategories of sports.

**Term weighting:** We use the VSM model \([60]\) to represent the collection. A document is represented as a point in multidimensional space by vector \( d_i = \{w_1, w_2, \ldots, w_v\} \), where \( w_j \) is the weight of a term \( t_j \) in the document. We use IDF \([51]\) as the weighting scheme that statistically weights the rare terms higher. We conjecture that IDF is more informative to differentiate a document from the collection, instead of the TF that weights common terms with higher weights. The weight of the term \( t_j \) is given as:

\[
w_j = idf_j = \log \left( \frac{|D|}{df_j} \right)
\]

where \( df_j \) is document frequency of term \( j \), the number of documents that contain the term. In this article, we calculate IDF after applying standard pre-processing steps to remove stop words and stemming.

**Nearest neighbors:** The cluster hypothesis proposed in IR \([29]\) states, “the linked sets of documents are likely to be relevant to the same request.” It led to prove theoretically by applying the reversed cluster hypothesis \([21]\) that “documents relevant to the same query should occur in the same cluster.” Prior research has empirically showed that when a document \( d_i \) is posed as a query to an IR system, the retrieved document set can be considered similar to \( d_i \) \([45, 53]\). In this article, we propose to use the top-\( m \) retrieved set as NNs of \( d_i \).

Let the document collection \( D \) be organized in the form of an Inverted indexed data structure stored in an IR system. Let \( d_i \in D \) be posed as a document query \( q \) using a set of distinct terms \( \{t_1, t_2, \ldots, t_s\} \) where \( s \leq v \) to the IR system. Given the query document \( q \), a ranking function \( R_f \) employed in the IR system returns the most relevant \( m \) document set, \( D_q \), as:

\[
R_f : q \rightarrow D_q = \{(d_p, r_p) : p = 1, 2, \ldots, m\}
\]

where the relevancy score \( r_p \) of a document \( d_p \) can be calculated as:

\[
\text{score}(q, d_p) = r_p = \sum_{t \in q} (\sqrt{t f_{t,d_p} \times idf_t^2 \times \text{norm}(t, d_p) })
\]
Table 2. Summary of the Proposed Algorithms

| Category              | Algorithm                                           | Concept          |
|-----------------------|-----------------------------------------------------|------------------|
| Rare Frequency based  | OIDF: Outlier Detection based on Inverse Document Frequency | ![Diagram](https://example.com/diagram.png) |
| Ranking based         | ORFS: Outlier Detection based on Ranking Function Score | ![Diagram](https://example.com/diagram.png) |
| k-occurrence based    | ORNC: Outlier Detection based on Ranked Neighborhood k-occurrences Count | ![Diagram](https://example.com/diagram.png) |

Definition 3.3 (Nearest Neighbors (NN)). A set of top-ranked documents retrieved by employing a ranking function $R_f$ in an IR system can be considered as NN of $d_i$.

In this article, we use the Elasticsearch search engine as the IR system and obtain top-$m$ ($m = 10$) documents as $k$-NN ($k = 10$) for each document in the collection. Prior research has shown that the precision at top-10 documents in the ranked list for a query is usually high, due to tight coupling with the topic and these top documents possess sufficient information richness [58]. When a document is posed as a query to the IR system, it is represented with the top-$s$ ($s = 10$) terms as in [53] ranked in the order of IDF. There exist several ranking functions such as $tf*idf$, BM25, BM25P and LM-JM [17, 53]. We propose to use the widely applied $tf*idf$ ranking function to measure the relevance between a document and a query as in Equation (3).

Reverse neighbors: This is the count of how often a data object appears in $k$-NNs of every other data object [48]. This is defined as retrievability of a document in IR literature [7]. A document that rarely appears in any other $k$-NNs will have a high chance of being an outlier. This can be considered as an alternative way to determine hubs of documents in the collection.

Definition 3.4 (k-Occurrences). The number of times $d_i \in D$ occurs in the $k$-NN set of other documents is defined as the number of $k$-occurrences of $d_i$. It is denoted as $N_k(d_i)$ and referred to as the reverse neighbor count of $d_i$.

Example: Consider the document collection in Figure 3(a) that is indexed in an IR system to obtain the set of all relevant results, as shown in Figure 6(a). The $k$-occurrences of document $d_1$ is $N_k(d_1) = 4$ in this collection as it appears in $k$-NN lists of documents $d_1$, $d_2$, $d_6$, and $d_9$. This is the reverse neighbor count of $d_1$.

Using these basic concepts and definitions, we propose three novel algorithms to identify outliers in a document collection as detailed in Table 2 and map them to the categories developed in the literature on (numerical) high-dimensional outlier detection. These methods capture outlier documents with rare terms with different document ranking techniques. (1) OIDF uses the average of the IDF weights of the terms to rank the documents. The IDF weighting scheme, that gives high importance to rare terms, can deal with high dimensional text representation to identify the outliers efficiently. (2) ORFS uses the IR ranking function to retrieve ranking scores for nearest neighbor documents and reciprocal of average similarity score is used to rank a document based on how dissimilar it is to nearest neighbors to identify the deviated documents. The use of scalable IR systems enables ORFS to deal with sparse larger text collections to identify a set of related documents as nearest neighbors in response to a given document following the cluster hypothesis in ORFS [30]. (3) ORNC extends this concept and uses the IR ranked responses to capture the
Outlier Detection in Text

3.2 Outlier Detection Based on Inverse Document Frequency: OIDF

Generally, IDF measures how much information a term carries in the collection and is able to differentiate the term as distinct in the collection. According to Equation (1), the IDF value of a rare term should be high. We conjecture that an outlier document will contain terms that deviate from the majority in the collection. We use the average IDF weight of a document combining all the terms as a measure to detect outliers. An outlier score $OS_{idf}$ is assigned to every document $d_i = \{w_1, w_2, \ldots, w_v\}$ that represents with weights of the included $v$ terms based on the average IDF weight as follows.

$$OS_{idf}^{d_i} = \frac{1}{|v|} \sum_{j=1}^{v} w_j$$

(4)

It is expected that the $OS_{idf}$ score, which captures rare term frequencies, will be high in outlier documents compared to inliers. This is explained in more detail in Appendix A. A document is defined as an outlier if the outlier score $OS_{idf}$ is greater than a control parameter $T_1$. We will discuss later in the experiment section, the setting of the control parameter systematically and automatically. Algorithm 1 in Figure 2 presents all the steps of the OIDF method.

Example: Consider the same example given in Figure 3(a) that consists of 11 documents related to 2 sports: Cricket and Rugby, and an outlier document. Figure 3(b) shows the IDF vector coefficients for each document with the average IDF value of each vector. It reveals that the average IDF value of the outlier - $d_{11}$ is much higher than the rest of the collection.

OIDF is a simple algorithm that identifies potential outlier candidates; however, it also generates a large number of false positives. We present two ranking-based algorithms that we propose to combine with OIDF to form ensemble approaches. These ensemble approaches can drastically reduce the search space for outliers and reduce false outliers (as evident in experiments).

3.3 Outlier Detection Based On Ranking Function Score: ORFS

The ranking concept can be used in outlier detection to assign an outlier score to observations based on the ranked list, assuming the observations at the top will get higher outlier scores. In this article, we propose to calculate a ranking score to each document using the IR system and assign
outlier scores. We assign an outlier score to each document using the relevancy scores generated through the IR system. As per Definition 3.3, an IR system uses a ranking function as shown in Equation (2) to determine the most relevant documents ranked by the relevancy score as calculated in Equation (3). The relevancy score represents the level of relevancy of a retrieved document to the query document as compared to the whole collection. We utilize the relevancy score, \( r_p \), of top-\( m \) (\( m = 10 \)) relevant documents for a given document \( d_i \) to show how consistent the given document is within the collection. The relevancy score of the document \( d_p \) to a query document (\( \text{score}(q, d_p) \) or \( r(p) \)) represents how similar \( d_p \) is to the query document. In contrast, the reciprocal of the relevancy score determines how much those two documents are dissimilar (i.e., deviated).

We propose to calculate the outlier score \( OS_r \) for a document \( d_i \in D \) as reciprocal of the average relevancy scores given by the search engine for top-10 relevant documents. It presents the degree of deviation as follows:

\[
OS_r^{d_i} = \frac{m}{\sum_{p=1}^{m} r_p} \left| d_i \neq d_p, \text{ where } r_p \geq 0 \right.
\]

A document is defined as an outlier if the outlier score \( OS_r \) is greater than the control parameter \( T_2 \).

### 3.3.1 Ensembles ORFS(I) and ORFS(S): Combining OIDF and ORFS

We propose to combine ORFS with OIDF to create an ensemble method to achieve robust outlier detection as in Figure 5. Previous researchers have built the ensemble models in two ways: independent and sequential ensembles [1]. We explore both approaches to reduce false positives.

Following the independent ensemble approach, both ORFS and OIDF algorithms generate the outlier candidates (i.e., \( D^f = D \)) and the common candidates in both sets have been identified as final outliers, as in Equation 6. This reduces the number of false outliers.

\[
D^o_f = D^o_{idf} \cap D^o_r
\]

Following the sequential ensemble approach, OIDF is first used to generate outlier candidates, then those candidates are tested through ORFS to calculate the outlier score \( OS_r \). A final set of outliers (\( D^o_f = D^o_r \)) is obtained using threshold \( T_2 \). This approach allows ORFS to search in a much smaller search space for outliers (i.e., \( D^f \subset D \)). Algorithm 2 in Figure 4 explains all the steps in this approach of relevancy score-based outlier detection. Combining the outliers generated by OIDF (that directly uses IDF weights) with the outliers generated by ORFS (that considers IDF with reciprocal retrieval score) reduces false detection as evident by the detailed experimental analysis later.
Example: Figure 6(a) shows the outlier scores calculated using ranking scores given by the Elasticsearch search engine for the same toy example. The outlier document \( d_{11} \) held the outlier score \( \gg 1 \) which highlighted it as the most possible outlier. As shown in Figure 3(b), OIDF also assigned the highest outlier score to \( d_{11} \) and made it the most suitable outlier candidate using both the independent and sequential ensemble approaches, after combining with ORFS.

3.4 Outlier Detection Based On Ranked Neighborhood \( k \)-Occurrences Count: ORNC

In high-dimensional data, hubs have been known to form local sub-dense neighborhoods instead of uniform distributions in a cluster [47]. We conjecture that outlier points would have less possibility to include in these hub regions and should have fewer \( k \)-occurrences in the nearest neighbor.
lists. In an indexed document collection, we obtain all sets of relevant documents using document queries to form initial search space. We use neighborhood documents calculated using Equation (2) with tf-idf function in Equation (3) for each document to obtain the lists of nearest neighbors. The $k$-occurrences count is measured within all the retrieved relevant documents (i.e., nearest neighbor sets) and used to define outlier scores based on the inverse of the count.

Let the documents retrieved in response to document query $d_i$ on $D$ be $D_{d_i}$ where $D_{d_i}$ is obtained using Equation (2). The outlier score $OS_c$ for $d_o \in D$ is calculated as:

$$OS_c^{d_o} = \frac{1}{|\{d_o \in D_{d_i}\}|}$$

Algorithm 3 in Figure 7 describes the overall process of ORNC where each document is assigned with an outlier score $OS_c$. If the score is greater than the control parameter $T_3$, the document is classified as an outlier.

3.4.1 Ensemble ORNC(S): Combining OIDF and ORNC. Identifying a neighborhood is an expensive operation due to the need for pair-wise comparisons [61]. Even in a smaller dataset, measuring
Outlier Detection in Text

Fig. 8. The ensemble approach of OIDF and ORNC for outlier detection.

$k$-occurrences by analyzing the nearest-neighbor list is highly expensive. Therefore, we propose to use only a selective set of outlier candidates to achieve effectiveness through the sequential ensemble approach. The initial set of outlier candidates is obtained using OIDF and the number of $k$-occurrences are measured, within all the retrieved relevant documents, for those candidate documents only. This sequential ensemble approach of OIDF with ORNC can identify the final outlier documents with reduced time and higher accuracy by reducing the search space, as in Figure 8.

Example: Figure 6(b) shows the outlier scores calculated using reverse neighbor count ($k$-occurrences) for all documents in the example document collection. The highest outlier score of 1 is given to the outlier candidate $d_{11}$ proposed by OIDF. It shows that the proposed method can identify the actual outlier.

In summary, the core concept used in three ensemble methods is the rare frequency-based outlier detection (OIDF). By using various IR ranking concepts, the quality of outlier detection of OIDF is improved by reducing false outliers. The ranking-based neighborhoods were used to provide outlier scores using pre-calculated relevancy scores in ORFS and $k$-occurrences in the response sets in ORNC.

4 EMPIRICAL ANALYSIS

In this section, we present the experimental evaluation of the proposed primary method OIDF and its ensemble approaches ORFS(I), ORFS(S), and ORNC(S) for accuracy, efficiency, and scalability. We conducted experiments on a single processor of 1.2 GHz Intel(R) Xeon(R) with a 264 GB shared memory. Algorithms were implemented in Python 3.5. Elasticsearch 2.4 was used as a search engine to provide relevant documents. First, we present the description of the real-world datasets used and the standard evaluation measures used to determine the accuracy of outlier detection. We show that the commonly used measures do not evaluate the outliers effectively; hence, we present a new evaluation criterion to report false predictions. The next few sections present the empirical analyses.

4.1 Datasets

Three categories of collections having documents of short, medium, and large length are used in experiments as shown by the column of average terms per document in Table 3. These datasets have ground-truth values that were used to measure the methods’ effectiveness extrinsically. These datasets are designed/selected to evaluate the performance of proposed methods against various challenges that exist on the text outlier detection problem: (1) different text vector
sizes; (2) different collection sizes; (3) the different number of classes; and (4) high vocabulary overlapping within inlier and outlier classes.

Our approach is distinct and more complex in comparison to existing methods as each collection contain multiple classes of documents—both inliers and outliers. Existing methods [31] do not include the diverse set of classes in their datasets that make the outlier detection process simpler and unnatural. They usually have one class of documents and outliers that do not belong to this class. We select a set of inlier and outlier classes, and attempt to identify outliers that are different from all these inlier classes which show less term overlapping with them.

DS1 contains inliers from multiple Wikipedia subcategories under “War” and outliers from 10 other categories not included inside “War.” DS2 contains inliers from five classes related to “Computers” and outliers from five other classes in 20Newsgroups. DS3 contains inliers from two classes and outliers from 25 other classes in the Reuters dataset. This dataset has classes with overlapping vocabulary showing a more complex scenario of outlier detection that has a considerably high number of overlapping terms between inlier and outlier classes. Inliers in short datasets (DS4 and DS5) are collected from classes that have at least 100 documents while two outliers per each class are collected from all other classes. These short datasets consist of more than 800 inlier and 400 outlier classes to explore the fine-grained scenarios as well as are large size. Table 3 shows a summary of these datasets.

| Datasets               | # of Docs | # of Unique Terms | # of Total Terms | # of Avg. Terms per doc | # of Outliers |
|------------------------|-----------|-------------------|------------------|-------------------------|---------------|
| Wikipedia (DS1)        | 11,521    | 305,827           | 9,206,250        | 799                     | 100           |
| 20News groups (DS2)   | 4,909     | 27,882            | 374,642          | 76                      | 50            |
| Reuters (DS3)         | 5,050     | 13,438            | 200,482          | 40                      | 50            |
| SED2013 (DS4)         | 81,228    | 46,548            | 1,583,073        | 19                      | 840           |
| SED2014 (DS5)         | 91,670    | 46,031            | 1,816,840        | 20                      | 976           |

4.2 Evaluation Measures

Accuracy is a well-known measure to define the effectiveness of outlier detection. Accuracy analyzes the percentage of correctness in predictions [26]. Let $TP$, $TN$, $FP$, $FN$ denote the correct outliers, correct inliers, incorrect outliers, and incorrect inliers respectively where $P$, $N$ denote the total number of outliers and inliers. The metric accuracy ($ACC$) is calculated as:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} = \frac{\text{total correct predictions}}{\text{total observations}} \quad (8)$$

However, the $ACC$ measure disregards the false outliers and false inliers. An explanation is provided in Appendix B.

Alternatively, the effectiveness of outlier detection is measured using the area under the Receiver Operating Characteristics (ROC) curve ($AUC$) [1, 40, 48]. The ROC curve shows the $TP$ rate ($TPR$) against $FP$ rate ($FPR$). Let $TPR$ and $FPR$ be:

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P} \quad (9)$$

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N} \quad (10)$$

ACM Transactions on Knowledge Discovery from Data, Vol. 14, No. 6, Article 71. Publication date: October 2020.
When \( T \) denotes a threshold to control outliers, \( AUC \) of \( ROC \) curve can be defined as:

\[
AUC = \int_0^1 ROC(T) \, dT
\]  
(11)

However, \( AUC \) also focuses only on correct predictions, which leads to a misleading picture of outlier detection without considering incorrect predictions. An explanation is provided in Appendix C. Consequently, there is a need for analyzing false positives and false negative predictions. The inverse of \( ACC \), which represents the total number of incorrect predictions against total observations, is not a clear measure of the false outlier and false inlier predictions. Some researchers have used \( FPR \) or \( FNR \) to report these. \( FPR \) considers false outliers (\( FP \)) against inliers in the dataset and shows linear variation within the range of 0 to 1 for the gradual increase of \( FP \). \( FNR \) considers false inliers (\( FN \)) against outliers in the dataset and shows linear variation within the range of 0 to 1 for the gradual increase of \( FN \). However, \( FPR \) and \( FNR \) provide relative values and are not able to differentiate the capability of a method to deal with false alarms.

We propose two measures Outlier Prediction Error (\( OPE \)) and Inlier Prediction Error (\( IPE \)) to emphasize false predictions over true predictions. \( OPE \) reports false outliers (\( FP \)) against true inliers (\( TN \)) and \( IPE \) reports false inliers (\( FN \)) against true outlier (\( TP \)).

\( OPE \) is defined as:

\[
OPE = \frac{FP}{TN + \epsilon} : \text{if } TN = 0 \text{ then } \epsilon = 1 \text{ else } \epsilon = 0
\]  
(12)

\( IPE \) is defined as:

\[
IPE = \frac{FN}{TP + \epsilon} : \text{if } TP = 0 \text{ then } \epsilon = 1 \text{ else } \epsilon = 0
\]  
(13)

The \( OPE \) (or \( IPE \)) measure varies in the range of 0 to \( N \) (or \( P \)) and can be divided into the following two ranges: 0 to 1; and 1 to \( N \) (or \( P \)). The value of \( OPE \) is within 0 to 1 if a method detects more true inliers than false outliers. However, a value greater than 1 indicates that more false outliers have been produced than true inliers. Similarly, a higher value of \( IPE \) than one shows a less effective method.

### 4.3 Baseline Algorithms

As primary baselines, we have chosen unsupervised outlier detection methods from the major categories in the existing literature to compare against the proposed unsupervised outlier detection methods. The benchmarking algorithms listed below are used as unsupervised baselines.

— Outlier detection using k-nearest neighbors (KNNO) [49]: This is a distance-based method where distance is calculated between each object and their \( k \)-NNs. Objects are then ranked based on the distance to \( k \)-NNs where top \( n \) objects are declared as outliers with user-defined \( n \). In this baseline method, we assign the number of outlier documents within each collection as 1% and 10 as the number of \( k \).

— Outlier detection using local density estimation (LOFO) [11]: This is a density-based method where a degree known as LOF is assigned to each object considering how isolated an object is related to its \( k \)-NNs (\( k \) is set as 10). LOF is defined as the ratio between the average densities of neighbors to the density of the object. Objects with high-rank LOF are defined as outliers. The threshold that governs the boundary between inliers and outliers is set as 1, in line with past research [8].

— Outlier detection using Non-Negative Matrix Factorization (NMFO) [31]: This is a recently developed matrix factorization-based approach specifically designed for text outlier detection. The \( l_2 \) norm assigned in the learning process of document-term matrix factorization is used as the outlier score for each document. The documents that get high-rank outlier
scores are defined as outliers. This method depends on several control parameters such as \(k, \alpha, \beta\). They are tuned to the best possible values after several parameter-tuning attempts. Best parameter values \(k, \alpha, \beta\) in DS2 and DS3 were set to \((20, 179, 0)\) and \((5, 23, 0)\), respectively, while DS1, DS4, and DS5 were set to \((20, 11, 0)\) following the description in [31] and yielding best results in multiple experimental settings.

—Mutual nearest-neighbor graph-based clustering method for detecting outliers (MNCO) [18]: This method is designed to cluster high dimensional sparse data by creating a considerably dense mutual neighbor graph. The points that do not belong to a cluster in the graph are considered noise or outliers. The two control parameters to define core dense regions in this baseline are set to 3 as they satisfy the minimum requirement, to be dense [45].

We also compare the TF-based IR ranking approach used for document similarity identification in proposed methods against the semantic embedding-based document similarity identification using doc2vec representation [38]. The set of similar documents and similarity scores for a document given by doc2vec are used in ORFS and ORNC algorithms to compare with IR-based ORFS and ORNC.

Neural network-based approaches have recently become popular in text mining with fully or weak supervision [35, 42, 43]. Although our proposed methods are fully unsupervised we have done experiments with a Convolutional neural network made for text classification [35] and a Generative adversarial active learning method for outlier detection [42]. Generative adversarial active learning for outlier detection [42] is a novel GAN-based semi-supervised approach used for outlier detection. A GAN model includes two networks where a generative network is used to generate candidates and a discriminator network is used to evaluate their validity [39, 41]. Although the GAN-based method [42] works in an unsupervised setting without relying on ground-truth labels of the data [32, 54], it follows a semi-supervised approach with active learning to generate initial outliers using the real data for the discriminator network. We follow the standard practice of using a dense word representation with reduced dimensionality obtained using the Global Vectors for Word Representation method [46] as the input to the neural networks in the experiments [35].

### 4.4 Accuracy Comparison

Accuracy of the proposed methods for large, medium and short text document collections is analyzed with the standard measures of \(\text{ACC}\), \(\text{ROC}\) curve, and \(\text{AUC}\) as well as the proposed measures of \(\text{OPE}\) and \(\text{IPE}\).

#### 4.4.1 Accuracy - \(\text{ACC}\)

In general, the proposed methods show improvement over the majority of baselines, especially when the dimensionality of the vector is high and the dataset is large. As detailed in Table 4, it is evident from the high \(\text{ACC}\) values of OIDF and its ensemble approaches that they outperformed all baselines except KNNO. However, it can be noted that KNNO is not scalable to high dimensional Wikipedia document collection (DS1). Moreover, KNNO requires the number of outlier documents as a control parameter which is a major limitation and makes it dependent on the parameter to achieve an improved outcome.

Within the proposed approaches, the basic OIDF algorithm yields the least accuracy as compared to ensemble methods, which can reduce false positives generated by OIDF. Ensembles methods, based on the IR ranking score, ORFS(I) and ORFS(S) perform similarly. As per \(\text{ACC}\), ORNC(S) is the best approach among the proposed methods. ORNC(S), based on the Hub concept, can achieve a higher level of performance even in extremely sparse short text data such as social media text (DS4, DS5). In the Reuters dataset (DS3), where classes in the collection are highly overlapping, it
Table 4. Accuracy Measure for Different Methods Against Datasets

| Dataset | Our Methods | Baseline Methods |
|---------|-------------|------------------|
|         | OIDF (I)    | ORFS (S)         | ORFS (S) | ORNC (S) | KNNO | LOFO | NMFO | MNCO |
| DS1     | 0.85        | 0.92             | 0.93     | 0.93     | *    | *    | *    | -    |
| DS2     | 0.87        | 0.93             | 0.94     | 0.94     | 0.99  | 0.01 | 0.98 | 0.06 |
| DS3     | 0.82        | 0.93             | 0.9      | 0.91     | 0.98  | 0.02 | 0.69 | 0.17 |
| DS4     | 0.82        | 0.9              | 0.9      | 0.93     | 0.99  | *    | 0.01 | -    |
| DS5     | 0.82        | 0.9              | 0.91     | 0.93     | 0.99  | *    | 0.01 | -    |
| Avg.    | 0.84        | 0.92             | 0.92     | 0.93     | 0.99  | 0.02 | 0.42 | 0.12 |

Note: (S), (I), “*”, and “-” denote the sequential ensemble approach, independent ensemble approach, aborted operations, and memory or runtime error, respectively.

Fig. 9. ROC curve and AUC for document collection with larger size term vectors (“*” denotes aborted operations, memory or runtime error).

It is hard to separate outliers considering terms in the VSM representation due to overlapping class behavior. This database yields relatively lower accuracy in most of the methods.

4.4.2 ROC and AUC. ACC considers the total correctness of predictions and does not provide a detailed analysis of true outliers and true inliers individually as compared to AUC (see Appendix B for more detail). As shown by graphs in Figures 9–11, OIDF provides the highest AUC values, except in short text datasets (DS4, DS5), due to its capacity to distinguish documents according to rare frequencies. This capacity helps OIDF to yield a higher TP rate and results in OIDF achieving higher AUC due to its separate analysis of TP (true outliers) and TN (true inliers), in contrast to ACC which represents total correct predictions. Both ranking score based ensemble methods, ORFS(I/S) perform similarly. The k-occurrences-based ensemble approach ORNC(S) outperformed the rest for short text data due to the hub-based concept that is specifically applicable to higher dimensions.

More specifically, Figure 9 shows the ROC curves of the Wikipedia document collection (DS1) derived by OIDF and its ensemble approaches. No baseline method could be executed on this dataset due to the large text size. This confirms the scalable nature of OIDF and its improved variations. Moreover, as seen by the results, the basic rare term-weight-based method OIDF has outperformed the ensemble methods. It states the power of a simple term weighting model in large documents to differentiate terms meaningfully where the occurrence of terms is considerably high within a document as well as in the respective collection. The IR ranking concept diluted the
effectiveness of this simple method, as depicted by Figure 9, by reducing the true outliers (TP) when the TP and TN (true inliers) are separately analyzed with AUC.

On the medium-sized collection (20News Group dataset (DS2) and Reuters dataset (DS3)), the proposed methods give higher AUC compared to baselines as shown in Figure 10. Similar to large text size collections, basic OIDF that simply considers the average rare frequency values of terms yields the highest AUC due to identifying the higher number of TPs. The KNNO method, which requires the number of outliers as a control parameter, is the best amongst other baselines for DS2 while NMFO performs as a random assignment. LOFO, which measures the density of a point relative to its’ neighbors density, shows the lowest performance in text data which is naturally sparse due to fewer term co-occurrences among documents. It could not differentiate the density around points in this sparse setting. LOFO identifies the majority of the inliers as outlier and results in lower ACC. MNCO, which uses a mutual nearest neighbor graph, outperforms other baselines in DS3 that contains overlapping class labels. In overlapping datasets, the ranking-based ORFS (I/S) methods and ORNC(S) yield a reduced performance in comparison to normal medium-sized datasets.

The ROC curves for document collections with short-term vectors are presented in Figure 11. These documents share a very few discriminative terms among similar documents compared to the other two dataset categories. Consequently, in this extremely sparse dataset, the ranking score-based ORNC(S) ensemble approach outperforms the basic OIDF method, due to the inclusion of a local sub-dense neighborhood (Hub) concept which is known to work for higher dimensions. Similarly, the distance-based KNNO, which uses pairwise distance difference comparison, does not perform well on this dataset, as on other datasets due to the distance concentration problem.

IR Ranked Similarity vs Semantic Similarity. Furthermore, we compare AUC results of ensemble ORFS(I/S) and ORNC(S) obtained using the IR ranking-based text similarity method against the
Table 5. AUC Comparisons Against Semantic Word Embedding-based Ranking

| Dataset | Our Methods | Baseline Methods with doc2vec similarity scores |
|---------|-------------|-----------------------------------------------|
|         | ORFS (I)    | ORFS (S) | ORNC (S) | ORFS with doc2vec | ORNC with doc2vec |
| DS1     | 0.70        | 0.69     | 0.71     | 0.72             | 0.56             |
| DS2     | 0.85        | 0.83     | 0.79     | 0.65             | 0.60             |
| DS3     | 0.7         | 0.69     | 0.7      | 0.67             | 0.60             |
| DS4     | 0.65        | 0.65     | 0.77     | 0.75             | 0.72             |
| DS5     | 0.65        | 0.65     | 0.78     | 0.75             | 0.72             |
| Avg.    | 0.71        | 0.70     | 0.75     | 0.71             | 0.64             |

Note: (S) and (I) denote the sequential ensemble approach.

Table 6. Performance Given by Neural Network-based Methods

| Dataset | Supervised CNN | GAN based Active Learning |
|---------|----------------|--------------------------|
|         | Accuracy-ACC   | Area Under the Curve- AUC |
| DS1     | 0.99           | 0.52                     |
| DS2     | 0.74           | 0.56                     |
| DS3     | 0.98           | 0.52                     |
| DS4     | 0.97           | 0.50                     |
| DS5     | 0.97           | 0.50                     |
| Avg.    | 0.93           | 0.52                     |

Performance of supervised/semi-supervised methods. Results in Table 6 shows the performance of neural network-based methods on the datasets. The convolutional neural network used for supervised text classification [35] provides almost same accuracy as ORNC(S) on average when compared with results in Table 4. Results show that except in DS2 that is small in collection size and with many classes, supervision based on training can provide superior results. This emphasizes the need of the presence of enough data to provide training in supervised methods. Results in Table 6 shows that it performs almost to a random method in sparse text data with weak supervision and inferior to the proposed fully unsupervised methods. Generally, these methods based on training a neural network are not effective and found time-consuming to use with the full-dimensional space.

4.4.3 Outlier Prediction Error and Inlier Prediction Error. To focus on false outliers and inliers, we next present the results with $\text{OPE}$ and $\text{IPE}$. Results in Table 7 support the conjecture that OIDF should be used as a basic method and the ranking-based algorithms ORFS or ORNC should be used to make an ensemble method with OIDF. ORFS, as a standalone method, generates a high level of
false outliers (i.e., \( OPE \) value is closer to 1 than 0) while giving more false inliers than OIDF on average. ORNC cannot be used as a basic method due to the high time complexity incurred by the increased number of comparisons with the size of the dataset.

As shown in Table 8, the ensemble methods ORNC(S) and ORFS (I/S) show a significant reduction in producing false outliers, making them suitable for real-world scenarios. The sequential ensemble approach ORNC(S) is successful in giving the fewest false outliers due to filtered candidates of outliers and becomes the best among the proposed methods in terms of \( OPE \). Furthermore, ORFS (I/S) also shows a reduction in false outliers. This confirms the importance of ensemble that reduces the false detection compared to OIDF that directly uses IDF weights for outlier detection or ORFS and ORNC that use IDF weights within the ranking function. KNNO shows good performance as it uses the specified number of outliers as an external input and obtains a controlled set of outlier documents. All other baseline methods generate a high amount of false outliers as well as some of them such as MNCO and LOFO fail to scale to large and high dimensional datasets. In this set-up, NMFO produces the worst performance due to the need for rigorous parameter tuning. The fine-grained nature of the large SED datasets (i.e., DS4 and DS5) impaired this process and we were unable to find realistic parameters even after a great effort.

Table 9 shows the inlier prediction error using \( IPE \). OIDF shows the least false inliers among the proposed methods. KNNO and NMFO become ineffective showing very high \( IPE \). LOFO and MNCO outperformed the proposed methods on the limited two datasets by producing lesser false inliers, however, they do not scale well for larger and high dimensional document collections.
Table 9. IPE for Different Methods Against Datasets (a Lower Value Near 0 is Better)

| Dataset | Our Methods | Baseline Methods |
|---------|-------------|------------------|
|         | OIDF        | ORFS (I) | ORFS (S) | ORNC (S) | KNNO | LOFO | NMFO | MNCO |
| DS1     | 0.47        | 1.13     | 1.17     | 1.08     | *    | *    | *    | -    |
| DS2     | 0.14        | 0.32     | 0.39     | 0.56     | 1.5  | 0    | 49   | 0    |
| DS3     | 0.61        | 1.17     | 1.08     | 1.08     | 49   | 0    | 1.63 | 0    |
| DS4     | 0.44        | 1.63     | 1.63     | 0.64     | 1    | *    | 0    | -    |
| DS5     | 0.38        | 1.48     | 1.58     | 0.6      | 0.94 | *    | 0    | -    |
| Avg.    | 0.41        | 1.15     | 1.17     | 0.79     | 13.1 | 0    | 12.66| 0    |

Note: (S), (I), “∗,” and “-” denote the sequential ensemble approach, independent ensemble approach, aborted operations, and memory or runtime error, respectively.

Fig. 12. Time and memory consumption for the proposed and benchmarking methods.

closer investigation on these two methods with OPE reveals that although they do not produce false inliers, they produce a larger portion of inliers as outliers (i.e., FP is extremely high). In contrast, the proposed methods produce a balanced performance with reduced false outliers and false inliers.

AUC and OPE together provide a complete picture of the effectiveness of proposed methods in outlier prediction. Further, IPE informs false inlier prediction which has been neglected in most of the outlier detection work.

4.5 Scalability and Computational Performance Analysis

Time taken by each method is shown in Figure 12(a). Results in this figure confirm that the proposed methods consume lesser time than the benchmarking methods in addition to the improved accuracy performance as shown in the previous sections. OIDF outperforms all the methods due to its simple rare document frequency-based calculation used for outlier filtering. Among the ensemble approaches, ORNC(S) shows the highest time consumption due to the requirement of a larger number of comparisons, though it can execute for all datasets due to a much smaller search space generated by the potential OIDF outliers. The independent ensemble approach in ORFS (i.e., ORFS (I)) shows a high time requirement due to additional iterations over the complete dataset.
The proposed methods outperformed all methods except on DS5 where the matrix factorization-based NMFO consumes slightly lesser time than ORNC(S). However, as shown in previous sections, NMFO produces inferior outcomes to ORNC(S). When data dimensionality is high, as in the Wikipedia dataset (DS1), the benchmarking methods were aborted due to exceptional high time consumption. Additionally, LOFO and MNCO could not handle the document collections with a large number of instances such as SED 2013 (DS4) and SED 2014 (DS5).

Figure 12(b) shows the memory consumption of each method. It shows that rare frequency-based OIDF and ranking scores based ORFS (I/S) ensemble approaches consume smaller memory in comparison to the ranked neighborhoods based ORNC(S) that considers \( k \)-occurrences similar to sub-dense local neighborhoods (Hubs) in high dimensionality. Figure 12(b) highlights that the proposed methods consume less memory, in comparison to baseline methods. KNNO, which achieves high accuracy in \( ACC \), shows higher memory and time consumption. Due to heavy time and memory requirements, all baseline methods are impaired when dealing with large term vectors such as Wikipedia and lead to resource starvation.

We further explore the scalability of the proposed methods considering incremental samples of SED 2013, which consists of short-term vectors. Figure 13 shows that all proposed methods have near-linear time complexity. The simplest rare frequency-based outlier detection OIDF shows the smallest time while the \( k \)-occurrences-based ensemble approach (i.e., ORNC(S)) consumes the highest time within our methods. Among the baseline methods, distance-based KNNO is the only method that scaled up to the largest sample we have used within the experiment. However, the success of KNNO depends on the number of outliers given as an external input and is not found successful with \( AUC \) which analyses inliers and outliers separately. Further, \( IPE \) which represents inlier prediction error is reported high for KNNO.

Table 10 shows the computational complexity of the proposed algorithms against baseline methods. Among the proposed algorithms, ORNC that defines the outlier score based on the number
of times a particular document appears in IR search results has the highest computational complexity. The sequential ensemble approach of this method, ORNC(S), cuts down this complexity by reducing the search space $n$. The baseline algorithms show comparatively higher computational complexity. This validates why LOFO and MNCO did not work for larger datasets while OIDF and ORFS worked efficiently.

### 4.6 Sensitivity Analysis

OIDF and its ensemble approaches use a threshold parameter similar to prior outlier detection algorithms [22, 49]. We set these control parameters automatically using the characteristics of the dataset, therefore, the proposed methods can be called parameter-free. All parameters that govern outlier scores have been explored considering intrinsic data characteristics such as mean, median and standard deviation. The control parameter $T_1$ of OIDF is set as the combination of median and standard deviation. The median that removes the effect of noise was boosted by adding standard deviation to detect the outliers that have a smaller portion within the document collections. It yields more true outliers as shown in Figure 14(a) for all the datasets except Reuters (DS3), which contains overlapping class labels for documents.

The control parameter $T_2$ in ORFS (I/S) and $T_3$ in ORNC(S) are set as the median value. Figure 14(b), (c) and (d) show how the quality of prediction varies amongst descriptive statistical measures. The median, which removes the unusual bursts in the outlier scores, gives the highest AUC except for DS3 in ORFS (S), which contains overlapping class labels.

An IR system employs different ranking functions such as LM Jelinek-Mercer Smoothing (LM-JM), LM Dirichlet Smoothing (LM-Dirichlet) and Okapi BM25 in addition to tf*idf [9]. However, LM-JM assigns negative scores to terms that have fewer occurrences and LM-Dirichlet captures important patterns in the text leaving the noise [17]. Therefore, they will be less effective in highlighting the outliers that have rare terms. In contrast, BM25 and tf*idf ranking functions show the capability to capture the deviated documents with rare terms using IDF of terms. Figure 15 shows...
the results provided by each proposed method with these two ranking functions. It shows that in general, tf*idf can more accurately identify outliers than the BM25 for the proposed methods. Therefore, we used tf*idf as the default ranking function.

4.7 Discussion
We summarize the interesting observations as follows:

— The basic algorithm OIDF, based on the simple concept of using rare terms in a document identified by the IDF scheme, shows high competence to detect deviations even for large document collections consuming less time. However, as shown by ACC and OPE results, OIDF is adversely affected by the higher number of false positives produced.

— The use of search engine ranking provides the advantage of obtaining relevant documents as similar documents from a large document collection for a document posed as a query. Reported results confirm the success of this approach in ensemble ORFS(I/S) and ORNC(S) methods.

— The higher accuracy achieved with ORNC(S) compared to OIDF and ORFS(I/S) can be attributed to identifying and using the sub-dense local neighborhoods present in higher dimensions. The count of \( k \)-occurrences allows identifying “hubs” and “anti-hubs,” which are away from local sub dense neighborhoods that have less \( k \)-occurrences count. These anti-hub points become probable outlier candidates. ORNC(S) even produces the best outcome for the short text size data where complexity is increased due to less term co-occurrence. However, it consumes substantial time due to the requirement of a large number of comparisons within each document neighborhood and will be less time-efficient for datasets with a larger number of documents.

— The strategy of combining different outlier detection approaches affects the effectiveness of outlier prediction. According to ACC and OPE measures, ORFS(I/S) and ORFS(S) outperformed the basic singleton OIDF method by reducing false positives. The improved time efficiency of ORFS(S), however, favors the sequential ensemble method compared to the independent ensemble method, as both produce nearly the same level of accuracy.

— While comparing with baselines, KNNO shows a higher ACC compared to the proposed approaches. The input parameter specifying the number of outliers is the reason behind this behavior. However, AUC that independently analyses the inlier and outlier prediction in detail confirms that KNNO reports comparatively higher false inliers. High memory consumption and false inlier prediction make KNNO a weak text outlier detection method.

— Reported results show that the state-of-the-art methods are not scalable to document collections with high term vectors. A mutual NN graph building process using \( k \)-NN calculation is not scalable to larger datasets due to the required high number of pairwise comparisons as evident with MNCO. NMFO is proposed to handle high dimensional term vectors through
Table 11. Applicability of the Proposed Methods

| Category                  | Method | Nature of the documents in outlier detection applications                                      | Functionality |
|---------------------------|--------|---------------------------------------------------------------------------------------------|---------------|
| Rare Frequency based      | OIDF   | Large text size collections such as Wikipedia                                               | Accuracy, Efficiency |
| Ranking based             | ORFS(S)| Medium text size collections such as newsgroup data                                           | Accuracy, Efficiency |
| Ranking based             | ORFS(I)| Medium text size collections such as newsgroup data                                           | Accuracy       |
| k-occurrence based        | ORNC(S)| Short text data such as social media which deal with extreme sparseness                     | Accuracy       |

a resulted error in dimensionality reduction. However, experiments with large term vectors (DS1) reveal that it cannot handle large text size collection. Additionally, experiments on datasets that consist of many groups (i.e., DS4 and DS5) show that the sum of the square error in non-negative matrix factorization is impaired in handling fine-grained problems, as evident by the OPE measure.

Finally, we summarize the proposed methods according to their suitability on different types of data in Table 11.

5 CONCLUSION

This article deals with the important topic of high-dimensional text outlier detection. In this data domain, the traditional distance or density-based outlier detection methods are challenged due to the distance concentration problem. Most of the state-of-the-art methods are impaired when the number of groups within a document collection is high, as it becomes difficult to generalize common patterns to identify deviation for outliers.

This article proposes a simple method of outlier detection based on the use of the IDF weighting scheme, OIDF. It effectively uses the notion of rare terms to identify the documents that deviate from the majority of documents in the collection. This method, however, suffers from generating high false positives and requires additional processing to improve accuracy. To handle efficacy and efficiency, we propose many ensemble approaches with OIDF using the ranking concept in IR systems, which has already been proven to handle high-dimensional larger document collections with reduced computational complexity. An IR system is used to retrieve the relevant documents for each document in the collection and the top-n relevant documents are considered to be the neighborhood of the document. ORFS uses the relevancy scores and ORNC uses the relevant document count to identify outliers.

We explore the most effective ensemble approach (i.e., independent or sequential) in combining ORFS and ORNC with OIDF. The sequential approach utilizes the outlier candidates identified by OIDF to reduce the search space for improving the quality of outlier detection. The ability of rare document frequency in identifying outliers in OIDF is enhanced by the IR concepts in ORFS and ORNC and reduces false positives compared to OIDF only. In the independent ensemble approach, both ORFS and OIDF algorithms generate the outlier candidates, and the common candidates in both sets have been identified to be the final outliers. The ORNC is not used in the independent approach to generate outliers due to high time complexity.
The empirical analysis is conducted on diverse datasets including large-, medium-, and short-term vector sizes with different numbers of classes and different levels of vocabulary overlapping. The proposed methods are benchmarked against several state-of-the-art distance-based, density-based, NMF-based, and graph-based outlier detection methods. Empirical analysis shows that the proposed methods are capable of detecting outliers in high-dimensional document collection with considerably high performance, including accuracy and efficiency. These approaches are designed in a threshold independent way by setting the control parameter autonomously based on the internal characteristics of the text collection.

This article presents a substantial work in the area of text outlier detection. However, identifying outliers in dynamic text streams with limited memory and time is important for novelty detection. Therefore, future directions are applying proposed algorithms on dynamic temporal text data for outlier and insight detection. Parallelizing these algorithms with possible improvement for run time and memory is also for our future investigation.

APPENDICES

A RATIONALE FOR THE OUTLIERS’ SCORE

Let \( OS_{idf}^{d_o} \) and \( OS_{idf}^{d_i} \) be the average of IDF values of all terms in an outlier document \( d_o \in D \) and inliner document \( d_i \in D \), respectively, in a document collection \( D \). We believe that \( OS_{idf}^{d_o} > OS_{idf}^{d_i} \) is valid for an outlier and inliner pair due to the following reasons.

- For a generic document \( d_k \in D \), IDF weight for a term can be calculated as in Equation (1) where rare terms get high IDF values due to their low document frequency (\( df \)) compared to common terms.
- An outlier document \( d_o \) with the average IDF weight of respective terms \( OS_{idf}^{d_o} \) calculated using Equation (4), will get higher value compared to the inliner document \( d_i \), as \( d_o \) consists of a set of rare terms within \( D \). It indicates a deviation from the majority.
- In contrast, an inliner document \( d_i \) will possess common terms that represent intrinsic themes of \( D \) and, thereby will hold a lower average IDF, \( OS_{idf}^{d_i} \), for respective terms.
- An \( OS_{idf}^{d_o} \), which is dominated by (rare) deviated terms should be higher than an \( OS_{idf}^{d_i} \), which is led by common terms within \( D \).

B WEAKNESS IN ACC MEASUREMENT

\( ACC \) measures the effectiveness of predictions in terms of correct predictions and does not consider false predictions. Consequently, it disregards the false outliers and false inliers.

\[
- ACC = \frac{TP + TN}{TP + FP + TN + FN} \\
- ACC = \frac{TP + TN}{P + N}
\]

\( ACC \) considers truly predicted instances against the total observations as a ratio and highlights only correct predictions. It can be considered a biased evaluation that neglects the incorrect predictions made by a method. Hence, 1-\( ACC \) can be used as an indirect indication of incorrect predictions, which represents total false predictions against total observations. However, \( ACC \) does not separately evaluate \( FP \) and \( FN \) that represent false outliers and false inliers to determine the error in outlier identification and inlier identification of a method.
C WEAKNESS IN AUC MEASUREMENT

AUC that considers a trade-off between TPR and FPR does not properly focus on false outliers and false inliers, and masks the false positives and false negatives. Let’s use a binary outlier detection scenario to produce the ROC curve as shown in Figure 16. AUC is the sum of the areas of A, B, and C.

\[
\text{AUC} = A + B + C \quad \text{as proved in [12]}
\]
\[
\text{AUC} = \frac{1}{2} \cdot TPR \cdot FPR + (1 - FPR) \cdot TPR + \frac{1}{2} \cdot (1 - TPR) \cdot (1 - FPR)
\]
\[
\text{AUC} = \frac{1}{2} \cdot TPR \cdot FPR + \frac{1}{2} \cdot (1 - FPR) \cdot (TPR + 1)
\]
\[
\text{AUC} = \frac{1}{2} \cdot (TPR + 1 - FPR)
\]
\[
\text{AUC} = \frac{1}{2} \left( \frac{TP}{TP + TN} + \frac{FP}{FP + TN} \right)
\]
\[
\text{AUC} = \frac{1}{2} \left( \frac{TP}{TP + TN} + \frac{TN}{FP + TN} \right)
\]

AUC informs true outliers out of total outliers and true inliers out of total inliers. Specifically, it details the correctly predicted outlier ratio and inlier ratio separately. It does not inform false outliers or false inliers efficiently as it does not treat false positives or false negatives with special care.

REFERENCES

[1] Charu C. Aggarwal. 2015. Outlier analysis. In Data Mining. Springer, 237–263.
[2] Charu C. Aggarwal and Philip S. Yu. 2001. Outlier detection for high dimensional data. In Proceedings of the ACM SIGMOD International Conference on Management of Data. Vol. 30. ACM, 37–46.
[3] Charu C. Aggarwal and ChengXiang Zhai. 2012. Mining Text Data. Springer Science & Business Media.
[4] Malik Agyemang, Ken Barker, and Rada S. Alhajj. 2005. WCOND-Mine: Algorithm for detecting web content outliers from Web documents. In Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC’05). IEEE, 885–890.
[5] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data. ACM, 49–60.
[6] Mazin Aouf and Laurence A. F. Park. 2012. Approximate document outlier detection using random spectral projection. In Proceedings of the Australasian Joint Conference on Artificial Intelligence. Springer, 579–590.
[7] Leif Azzopardi and Vishwa Vinay. 2008. Retrievability: An evaluation measure for higher order information access tasks. In Proceedings of the 17th ACM Conference on Information and Knowledge Management. ACM, 561–570.
[8] L. Douglas Baker, Thomas Hofmann, Andrew McCallum, and Yiming Yang. 1999. A hierarchical probabilistic model for novelty detection in text. In Proceedings of International Conference on Machine Learning.
[9] Graham Bennett, Falk Scholer, and Alexandra Uidtenbogerd. 2008. A comparative study of probabilistic and language models for information retrieval. In Proceedings of the 19th Conference on Australasian Database - Volume 75. Australian Computer Society, Inc., 65–74.

[10] Leonid Blouvshtein and Daniel Cohen-Or. 2018. Outlier detection for robust multi-dimensional scaling. IEEE Transactions on Pattern Analysis and Machine Intelligence 49, 9 (2018), 2273–2279.

[11] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Vol. 29. ACM, 93–104.

[12] Scott B. Cantor and Michael W. Kattan. 2000. Determining the area under the ROC curve for a binary diagnostic test. Medical Decision Making 20, 4 (2000), 468–470.

[13] Nick Cercone, Farzana Yasmeen, and Yasser Gonzalez-Fernandez. 2014. Information Retrieval and the Vector Space Model. University Lecture.

[14] Debasrita Chakraborty, Vaasudev Narayanan, and Ashish Ghosh. 2019. Integration of deep feature extraction and ensemble learning for outlier detection. Pattern Recognition 89 (2019), 161–171.

[15] Joseph DiGrazia, Karissa McKelvey, Johan Bollen, and Fabio Rojas. 2013. More tweets, more votes: Social media as a quantitative indicator of political behavior. PloS One 8, 11 (2013), e79949.

[16] Lian Duan, Lida Xu, Ying Liu, and Jun Lee. 2009. Cluster-based outlier detection. Annals of Operations Research 168, 1 (2009), 151–168.

[17] Elasticsearch. 2019. Similarity Module. Retrieved from https://www.elastic.co/guide/en/elasticsearch/reference/master/index-modules-similarity.html

[18] Levent Eröz, Michael Steinbach, and Vinip Kumar. 2003. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In Proceedings of the 2003 SIAM International Conference on Data Mining. SIAM, 47–58.

[19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Vol. 96. 226–231.

[20] Arthur Flexer. 2016. Hubness-aware outlier detection for music genre recognition. In Proceedings of the 19th International Conference on Digital Audio Effects.

[21] Norbert Fuhr, Marc Lechtenfeld, Benno Stein, and Tim Gollub. 2012. The optimum clustering framework: Implementing the cluster hypothesis. Information Retrieval 15, 2 (2012), 93–115.

[22] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti. 2004. Outlier detection using k-nearest neighbour graph. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04). Vol. 3. IEEE, 430–433.

[23] Douglas M. Hawkins. 1980. Identification of Outliers. Vol. 11. Springer.

[24] Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. Pattern Recognition Letters 24, 9–10 (2003), 1641–1650.

[25] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. 2005. A brief survey of text mining. In Ldv Forum, Vol. 20. Citeseer, 19–62.

[26] Jinlong Huang, Qingsheng Zhu, Lijun Yang, and Ji Feng. 2016. A non-parameter outlier detection algorithm based on natural neighbor. Knowledge Based Systems 92 (2016), 71–77.

[27] IBM. 2017. Big Data and Analytics Hub. Retrieved from https://www.ibmbigdatahub.com/blog/what-text-analytics.

[28] Donald A. Jackson and Yong Chen. 2004. Robust principal component analysis and outlier detection with ecological data. Environmetrics: The Official Journal of the International Environmetrics Society 15, 2 (2004), 129–139.

[29] Nick Jardine and Cornelis Joost van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. Information Storage and Retrieval 7, 5 (1971), 217–240.

[30] Raymond Austin Jarvis and Edward A. Patrick. 1973. Clustering using a similarity measure based on shared near neighbors. IEEE Transactions on Computers 100, 11 (1973), 1025–1034.

[31] Ramakrishnan Kannan, Hyenkyn Woo, Charu C. Aggarwal, and Haesun Park. 2017. Outlier detection for text data. In Proceedings of the 2017 SIAM International Conference on Data Mining. SIAM, 489–497.

[32] Pei Ke, Fei Huang, Minjie Huang, and Xiaoyan Zhu. 2019. ARAML: A stable adversarial training framework for text generation. Arxiv Preprint Arxiv:1908.07195 (2019).

[33] Edwin M. Knox and Raymond T. Ng. 1998. Algorithms for mining distance-based outliers in large datasets. In Proceedings of the International Conference on Very Large Data Bases. 392–403.

[34] Samrat Kokkula and Narasimha Murty Musti. 2013. Classification and outlier detection based on topic based pattern synthesis. In Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer, 99–114.

[35] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. Information 10, 4 (2019), 150.
Outlier Detection in Text

[36] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. 2009. Outlier detection in axis-parallel subspaces of high dimensional data. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 831–838.

[37] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. 2008. Angle-based outlier detection in high-dimensional data. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 444–452.

[38] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning. 1188–1196.

[39] Shangsong Liang. 2019. Unsupervised semantic generative adversarial networks for expert retrieval. In Proceedings of the World Wide Web Conference. ACM, 1039–1050.

[40] Huawen Liu, Xuelong Li, Jiuyong Li, and Shichao Zhang. 2017. Efficient outlier detection for high-dimensional data. IEEE Transactions on Systems, Man, and Cybernetics: Systems 48, 12 (2017), 2451–2461.

[41] Lining Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li. 2018. Generative adversarial network for abstractive text summarization. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence.

[42] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jiashan Sun, Meng Wang, and Xiangnan He. 2020. Generative adversarial active learning for unsupervised outlier detection. IEEE Transactions on Knowledge and Data Engineering 32, 8 (2020), 1517–1528.

[43] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. 6826–6833.

[44] Wathsala Anupama Mohotti and Richi Nayak. 2018. Corpus-based augmented media posts with density-based clustering for community detection. In Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI’18). IEEE, 379–386.

[45] Wathsala Anupama Mohotti and Richi Nayak. 2018. An efficient ranking-centered density-based document clustering method. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 439–451.

[46] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP’14). 1532–1543.

[47] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research 11, Sep (2010), 2487–2531.

[48] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2014. Reverse nearest neighbors in unsupervised distance-based outlier detection. IEEE Transactions on Knowledge and Data Engineering 27, 5 (2014), 1369–1382.

[49] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In Proceedings of the ACM SIGMOD International Conference on Management of Data. Vol. 29. ACM, 427–438.

[50] Thomas Roelleke and Jun Wang. 2008. TF-IDF uncovered: A study of theories and probabilities. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 435–442.

[51] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24, 5 (1988), 513–523.

[52] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. 2015. Fast and scalable outlier detection with approximate nearest neighbor ensembles. In Proceedings of the International Conference on Database Systems for Advanced Applications. Springer, 19–36.

[53] Taufik Sutanto and Richi Nayak. 2015. Semi-supervised document clustering via loci. In Proceedings of the International Conference on Web Information Systems Engineering. Springer, 208–215.

[54] Rui Wang, Deyu Zhou, and Yulan He. 2019. Open event extraction from online text using a generative adversarial network. Arxiv Preprint Arxiv:1908.09246 (2019).

[55] S. K. Michael Wong, Wojciech Ziarko, and Patrick C. N. Wong. 1985. Generalized vector spaces model in information retrieval. In Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 18–25.

[56] Yiming Yan, Ruizhang Huang, Can Ma, Liyang Xu, Zhiyuan Ding, Rui Wang, Ting Huang, and Bowei Liu. 2017. Improving document clustering for short texts by long documents via a dirichlet multinomial allocation model. In Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web Information Retrieval and Knowledge Management. Springer, 626–641.

[57] Zhong Yuan, Xianyong Zhang, and Shan Feng. 2018. Hybrid data-driven outlier detection based on neighborhood information entropy and its developmental measures. Expert Systems with Applications 112 (2018), 243–257.

[58] Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. 2005. Improving web search results using affinity graph. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 504–511.

[59] Jiangong Zhang, Xiaohui Long, and Torsten Suel. 2008. Performance of compressed inverted list caching in search engines. In Proceedings of the 17th International Conference on World Wide Web. ACM, 387–396.
[60] Weizhong Zhao, Qing He, Huifang Ma, and Zhongzhi Shi. 2012. Effective semi-supervised document clustering via active learning with instance-level constraints. *Knowledge and Information Systems* 30, 3 (2012), 569–587.

[61] Pingfei Zhu, Xiangwen Zhan, and Wenming Qiu. 2015. Efficient k-nearest neighbors search in high dimensions using mapreduce. In *Proceedings of the 2015 IEEE 5th International Conference on Big Data and Cloud Computing*. IEEE, 23–30.

[62] Arthur Zimek. 2018. Clustering high-dimensional data. In *Data Clustering*. Chapman and Hall/CRC, 201–230.

Received May 2018; revised April 2020; accepted May 2020