Knowl Inf Syst (2013) 36:1–21
DOI 10.1007/s10115-012-0552-3

REGULAR PAPER

# Towards graphical models for text processing

**Charu C. Aggarwal · Peixiang Zhao**

Received: 29 November 2011 / Revised: 29 June 2012 / Accepted: 22 August 2012 /
Published online: 13 September 2012


**Abstract** The rapid proliferation of the World Wide Web has increased the importance and prevalence of text as a medium for dissemination of information. A variety of text mining and management algorithms have been developed in recent years such as clustering, classification, indexing, and similarity search. Almost all these applications use the well-known *vector-space model* for text representation and analysis. While the vector-space model has proven itself to be an effective and efficient representation for mining purposes, it does not preserve information about the ordering of the words in the representation. In this paper, we will introduce the concept of *distance graph representations* of text data. Such representations preserve information about the relative ordering and distance between the words in the graphs and provide a much richer representation in terms of sentence structure of the underlying data. Recent advances in graph mining and hardware capabilities of modern computers enable us to process more complex representations of text. We will see that such an approach has clear advantages from a qualitative perspective. This approach enables knowledge discovery from text which is not possible with the use of a pure vector-space representation, because it loses much less information about the ordering of the underlying words. Furthermore, this representation does not require the development of new mining and management techniques. This is because the technique can also be converted into a structural version of the vector-space representation, which allows the use of *all existing tools for text*. In addition, existing techniques for graph and XML data can be directly leveraged with this new representation. Thus, a much wider spectrum of algorithms is available for processing this representation. We will apply this technique to a variety of mining and management applications and show its advantages and richness in exploring the structure of the underlying text documents.

**Keywords** Text clustering · Text classification · Text representation · Text search

This paper is an extended version of Aggarwal and Zhao [3].

C. C. Aggarwal (✉)
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
e-mail: charu@us.ibm.com

P. Zhao
Florida State University, Tallahassee, FL 32306, USA
e-mail: zhao@cs.fsu.edu

## 1 Introduction

Text management and mining algorithms have seen increasing interest in recent years, because of a variety of Internet applications such as the World Wide Web, social networks, and the blogosphere. In its most general form, text data can be represented as strings, though simplified representations are used for effective processing. The most common representation for text is the *vector-space representation* [20]. The vector-space representation treats each document as an unordered "bag of words". While the vector-space representation is very efficient because of its simplicity, it loses information about the structural ordering of the words in the document, when used purely in the form of individual word representations.

For many applications, the "unordered bag of words" representation is not sufficient for the purpose of analytical insights. This is especially the case for fine-grained applications in which the structure of the document plays a key role in the underlying semantics. One advantage of the vector-space representation is that its simplicity lends itself to straightforward processing. The efficiency of the vector-space representation has been a key reason that it has remained the technique of choice for a variety of text processing applications. On the other hand, the vector-space representation is *very lossy* because it contains absolutely no information about the ordering of the words in the document. One of the goals of this paper is to design a representation which retains at least some of the ordering information among the words in the document without losing its flexibility and efficiency for data processing.

While the processing-efficiency constraint has remained a strait-jacket on the development of richer representations of text, this constraint has become easier to overcome in recent years because of a variety of hardware and software advances:

- The computational power and memory of desktop machines have increased by more than an order of magnitude over the last decade. Therefore, it has become increasingly feasible to work with more complex representations.
- The database community has seen tremendous algorithmic and software advances in management and mining of a variety of structural representations such as graphs and XML data [1]. In the last decade, a massive infrastructure has been built around mining and management applications for structural and graph data such as indexing [24,27,28,31,33], clustering [5], and classification [30]. This infrastructure can be leveraged with the use of structural representations of text.

In this paper, we will design graphical models for representing and processing text data. In particular, we will define the concept of *distance graphs*, which represents the document in terms of the distances between the distinct words. We will then explore a few mining and management applications with the use of the structural representation. We will show that such a representation allows for more effective processing and results in high-quality representations. This can retain rich information about the behavior of the underlying data. This richer level of structural information can provide two advantages. First, it enables applications which are not possible with the more lossy vector-space representation. Second, the richer representation provides higher quality results with existing applications. In fact, we will see that the only additional work required is a change in the underlying representation, and *all existing text applications can be directly used* with a vector-space representation of the structured data. We will present experimental results on a number of real data sets illustrating the effectiveness of the approach.

This paper is organized as follows. In the next section, we will explore the concept of distance graphs, and some of the properties of the resulting graphs. In Sect. 3, we will show how

to leverage distance graphs for a variety of mining and management applications. Section 4 discusses the experimental results. The conclusions and summary are presented in Sect. 5.

## 2 Distance graphs

In this section, we will introduce the concept of distance graphs, a graphical paradigm which turns out to be an effective text representation for processing. While the vector-space representation maintains no information about the ordering of the words, the string representation is at the other end of spectrum in maintaining all ordering information. Distance graphs are a natural intermediate representation which preserve a high level of information about the ordering and distance between the words in the document. At the same time, the structural representation of distance graphs makes it an effective representation for text processing. Distance graphs can be defined to be of a variety of *orders* depending upon the level of distance information which is retained. Specifically, distance graphs of order $k$ retain information about word pairs which are at a distance of at most $k$ in the underlying document. We define a distance graph as follows:

**Definition 2.1** A distance graph of order $k$ for a document $D$ drawn from a corpus $\mathcal{C}$ is defined as graph $G(\mathcal{C}, D, k) = (N(\mathcal{C}), A(D, k))$, where $N(\mathcal{C})$ is the set of nodes defined specific to the corpus $\mathcal{C}$, and $A(D, k)$ is the set of edges in the document. The sets $N(\mathcal{C})$ and $A(D, k)$ are defined as follows:

- The set $N(\mathcal{C})$ contains one node for each distinct word in the entire document corpus $\mathcal{C}$. Therefore, we will use the term "node $i$" and "word $i$" interchangeably to represent the index of the corresponding word in the corpus. Note that the corpus $\mathcal{C}$ may contain a large number of documents, and the index of the corresponding word (node) remains unchanged over the representation of the different documents in $\mathcal{C}$. Therefore, the set of nodes is denoted by $N(\mathcal{C})$ and is a function of the corpus $\mathcal{C}$.
- The set $A(D, k)$ contains a directed edge from node $i$ to node $j$ if the word $i$ precedes word $j$ by **at most** $k$ positions. For example, for successive words, the value of $k$ is 1. The frequency of the edge is the number of times that word $i$ precedes word $j$ by at most $k$ positions in the document.

We note that the set $A(D, k)$ always contains an edge from each node to itself. The frequency of the edge is the number of times that the word precedes itself in the document at a distance of at most $k$. Since any word precedes itself at distance 0 by definition, the frequency of the edge is **at least** equal to the frequency of the corresponding word in the document.

Most text collections contain many frequently occurring words such as prepositions, articles, and conjunctions. These are known as *stop-words*. Such words are typically not included in vector-space representations of text. Similarly, for the case of the distance graph representation, it is assumed that these words are removed from the text *before* the distance graph construction. In other words, stop-words are not counted while computing the distances for the graph and are also not included in the node set $N(\mathcal{C})$. This greatly reduces the number of edges in the distance graph representation. This also translates to better efficiency during processing.

We note that the order-0 representation contains only self-loops with corresponding word frequencies. Therefore, this representation is quite similar to the vector-space representation. Representations of different orders represent insights about words at different distances in the document. An example of the distance graph representation for a well-known nursery rhyme

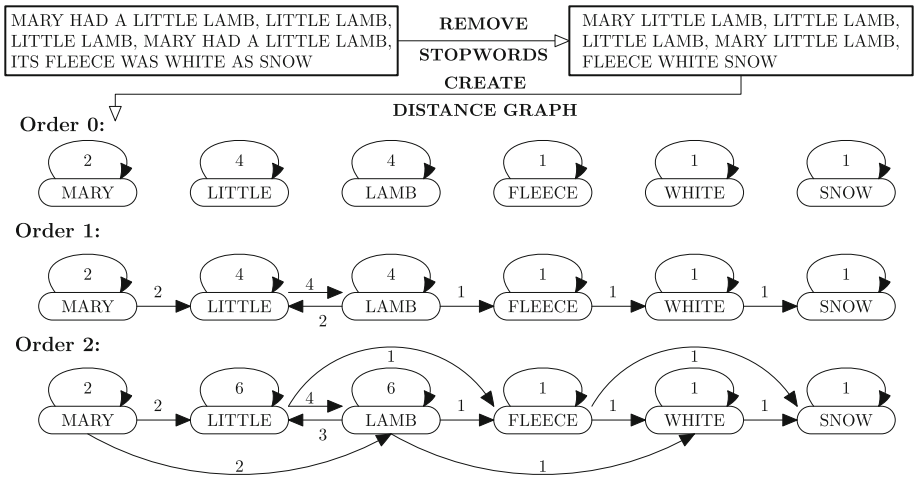


**Fig. 1** Illustration of distance graph representation

"Mary had a little lamb" is illustrated in Fig. 1. In this figure, we have illustrated the distance graphs of orders 0, 1, and 2 for the text fragment. The distance graph is constructed only with respect to the remaining words in the document, after the stop-words have already been pruned. The distances are then computed with respect to the pruned representation. Note that the distance graphs of order 0 contain only self-loops. The frequencies of these self-loops in the order-0 representation correspond to the frequency of the word, since this is also the number of times that a word occurs within a distance of 0 of itself. The number of edges in the representation will increase for distance graphs of successively higher orders. Another observation is that the frequency of the self-loops in distance graphs of order 2 increases over the order-0 and order-1 representations. This is because of repetitive words like "little" and "lamb" which occur within alternate positions of one another. Such repetitions do not change the frequencies of order-0 and order-1 distance graphs, but do affect the order-2 distance graphs. We note that distance graphs of higher orders may sometimes be richer, though this is not necessarily true for orders higher than 5 or 10. For example, a distance graph with order greater than the number of distinct words in the document would be a complete clique. Clearly, this does not necessarily encode useful information. On the other hand, distance graphs of order-0 do not encode a lot of useful information either. In the experimental section, we will examine the relative behavior of the distance graphs of different orders and show that distance graphs of low orders turn out to be the most effective.

From a database perspective, such distance graphs can also be represented in XML with attribute labels on the nodes corresponding to word-identifiers, and labels on the edges corresponding to the frequencies of the corresponding edges. Such a representation has the advantage that numerous data management and mining techniques for semi-structured data have already been developed. These can directly be used for such applications. Distance graphs provide a much richer representation for storage and retrieval purposes, because they partially store the structural behavior of the underlying text data. In the next section, we will discuss some common text applications such as clustering, classification, and frequent pattern mining and show that these problems can easily be solved with the use of the distance graph representation.

An important characteristic of distance graphs is that they are relatively sparse and contain a small number of edges for low values of the order $k$. As we will see in the experimental

section, it suffices to use low values of $k$ for effective processing in most mining applications. We make the following observations about the distance graph representation:

**Observation 2.1** Let $f(D)$ denote the number of words[1] in document $D$ (counting repetitions), of which $n(D)$ are distinct. Distance graphs of order $k$ contain **at least** $n(D) \cdot (k+1) - k \cdot (k-1)/2$ edges, and **at most** $f(D) \cdot (k+1)$ edges.

The above observation is simple to verify, since each node (except possibly for nodes corresponding to the last $k$ words) contains a self-loop along with at least $k$ edges. In the event that the word occurs multiple times in the document, the number of edges out of a node may be larger than $k$. Therefore, if we do not account for the special behavior of the last $k$ words in the document, the number of edges in a distance graph of order $k$ is at least $n(D) \cdot (k+1)$. By accounting for the behavior of the last $k$ words, we can reduce the number of edges by *at most* $k \cdot (k-1)/2$. Therefore, the total number of edges is given by *at least* $n(D) \cdot (k+1) - k \cdot (k-1)/2$. Furthermore, the sum of the outgoing frequencies from the different nodes is exactly $f(D) \cdot (k+1) - k \cdot (k-1)/2$. Since each edge has frequency at least 1, it follows that the number of edges in the graph is at most $f(D) \cdot (k+1)$. In practice, the storage requirement is much lower because of repetitions of word occurrences in the document. The modest size of the distance graph is extremely important from the perspective of storage and processing. In fact, the above observation suggests that for small values of $k$, the total storage requirement is not much higher than that required for the vector-space representation. This is a modest price to pay for the syntactic richness captured by the distance graph representation. We first make an observation for documents of a particular type, namely documents which contain only distinct words.

**Observation 2.2** Distance graphs of order 2 or less, which correspond to documents containing only distinct words, are planar.

The above observation is straightforward for graphs of order 0 (self-loops) and order 1 (edges between successive words). Next, we verify this observation for graphs of order 2. Note that if the document contains only distinct words, then we can represent the nodes in a straight line, corresponding to the order of the words in the document. The distinctness of words ensures that all edges are to nodes one and two positions ahead, and there are no backward edges. The edges emanating from odd numbered words can be positioned *above* the nodes, and the edges emanating from even numbered words can be positioned below the nodes. It is easy to see that no edges will cross in this arrangement.

In practice, documents are likely to contain non-distinct words. However, the frequencies are usually quite small, once the stop-words have been removed. This means that the graph tends to approximately satisfy the pre-conditions of Observation 2.2. This suggests that lower-order distance graph representations of most documents are either planar or approximately planar. This property is useful since we can process planar graphs much more efficiently for a variety of applications. Even for cases in which the graphs are not perfectly planar, one can use the corresponding planar algorithms in order to obtain extremely accurate results.

We note that the distance graphs are somewhat related to the concept of using $n$-grams for text mining [8,9]. However, $n$-grams are typically mined a priori based on their relative frequency in the documents. Such $n$-grams represent only a small fraction of the structural relationships in the document and are typically not representative of the overall structure in the document. A related area of research is that of collocation processing [13,17,22]. In collocation processing, the frequent sequential patterns of text are used to model word

[1] We assume that stop-words have already been removed from the document $D$.

dependencies, and these are leveraged for the purpose of online processing. As in the case of $n$-grams, collocation processing is only concerned with *aggregate* patterns in a collection, rather than the representation of a single text document with the precise ordering of the words in it. This can lead to a number of differences in the capabilities of these techniques. For example, in the case of a similarity search application, methods such as collocation processing may often miss many specific sequential patterns of words which occur in common between a pair of documents, if they do not occur frequently throughout the collection. Stated simply, the distance graph is a representation for text, which is *independent of the aggregate patterns in the other documents of the collection*.

Next, we examine the structural properties of documents which are reflected in the distance graphs. These structural properties can be leveraged to perform effective mining and management of the underlying data. A key structural property retained by distance graphs is that it can be used to detect identical portions of text shared by the two documents. This can be useful as a sub-task for a variety of applications (e.g., detection of plagiarisms) and cannot be achieved with the use of the vector-space model. Thus, the distance graph representation provides *additional functionality which is not available with the vector-space model*. We summarize this property as follows:

**Observation 2.3** Let $D_1$ and $D_2$ be two documents from corpus $\mathcal{C}$ such that the document $D_1$ is a subset of document $D_2$. Then, the distance graph $G(\mathcal{C}, D_1, k)$ is a subgraph of the distance graph $G(\mathcal{C}, D_1, k)$.

While the reverse is not *always* true (because of repetition of words in a document), it is *often* true because of the complexity of the text structure captured by the distance graph representation. This property is extremely useful for *retrieval by precise text fragment*, since subgraph-based indexing methods are well known in the graph and XML processing literature [24,26–31,33]. Thus, subgraph-based retrieval methods may be used to determine a close superset of the required document set. This is a much more effective solution than that allowed by the vector-space representation, since the latter only allows indexing by word membership rather than precise-sentence fragments.

Observation 2.3 can be easily generalized to the case when the two documents share text fragments without a direct subset relationship:

**Observation 2.4** Let $D_1$ and $D_2$ be two documents from corpus $\mathcal{C}$ such that they share the contiguous text fragment denoted by $F$. Then, the distance graphs $G(\mathcal{C}, D_1, k)$ and $G(\mathcal{C}, D_2, k)$ share the subgraph $G(\mathcal{C}, F, k)$.

We further note that a fragment $F$ corresponding to a contiguous text fragment will always be connected. Of course, not all connected subgraphs correspond to contiguous text fragments, though this may often be the case for the smaller subgraphs. The above observation suggests that by finding frequent connected subgraphs in a collection, it may be possible to determine an effective mapping to the frequent text fragments in the collection. A number of efficient algorithms for finding such frequent subgraphs have been proposed in the database literature [26,29]. In fact, this approach can be leveraged directly to determine possible plagiarisms (or commonly occurring text fragments) in very large document collections. We note that this would not have been possible with the "bag of words" approach of the vector-space representation, because of the loss in the underlying word-ordering information.

It is also possible to use the technique to determine documents such that some local part of this document discusses a particular topic. It is assumed that this topic can be characterized by a set $S$ of $m$ closely connected keywords. In order to determine such documents, we first synthetically construct a *bi-directional directed* clique containing these $m$ keywords (nodes).

A bi-directional directed clique is one in which edges exist in both directions for every pair of nodes. In addition, it contains a single self-loop for every node. Then, the *aggregate frequency of the edge-wise intersection* of the clique with the graph $G(\mathcal{C}, D, k)$ represents the number of times that the corresponding keywords occur within distance[2] of at most $k$ with one another in the document. This provides an idea of the local behavior of the topics discussed in the collection.

**Observation 2.5** Let $F_1$ be a bi-directional clique containing $m$ nodes and $D$ be a document from corpus $\mathcal{C}$. Let $E$ be the **edge-wise intersection** of the set of edges from $G(\mathcal{C}, D, k)$ which are contained with those in $F_1$. Let $q$ be the sum of the frequency of the edges in $E$. Then, $q$ represents the number of times that the keywords in the nodes corresponding to $F_1$ occur within a distance of $k$ of one another in the document.

The above property could also be used to determine documents which contain different topics discussed in different parts of it. Let $S_1$ and $S_2$ be the sets of keywords corresponding to two different topics. Let $F_1$ and $F_2$ be two bi-directional cliques corresponding to sets $S_1$ and $S_2$, respectively. Let $F_{12}$ be a bi-directional clique containing the nodes in $S_1 \cup S_2$. Similarly, let $E_1(D)$, $E_2(D)$ and $E_{12}(D)$ be the intersections of the edges of $G(\mathcal{C}, D, k)$ with $F_1$, $F_2$ and $F_{12}$, respectively. Note that the set of edges in $E_{12}(D)$ is a superset of the edges in $E_1(D) \cup E_2(D)$. Intuitively, the topics corresponding to $F_1$ and $F_2$ are discussed in different parts of the document if the frequencies of the edges in $E_1(D)$ and $E_2(D)$ are large, but the frequency of the edges in $E_{12}(D) - (E_1(D) \cup E_2(D))$ is extremely low. Thus, we can formalize the problem of determining whether a document contains the topics corresponding to $S_1$ and $S_2$ discussed in different parts of it.

**Formal Problem Statement**: *Determine all documents D such that the frequency of the edges in* $E_1(D) \cup E_2(D)$ *is larger than* $s_1$. *but the frequency of the edges in* $E_{12}(D) - (E_1(D) \cup E_2(D))$ *is less than* $s_2$.

## 2.1 The undirected variation

Note that the distance graph is a directed graph, since it accounts for the ordering of the words in it. In many applications, it may be useful to relax the ordering a little bit in order to allow some flexibility in the distance graph representation. Furthermore, undirected graphs allow for a larger variation of the number of applications that can be used, since they are much simpler to deal with for mining applications.

**Definition 2.2** An undirected distance graph of order $k$ for a document $D$ drawn from a corpus $\mathcal{C}$ is defined as graph $G(\mathcal{C}, D, k) = (N(\mathcal{C}), A(D, k))$, where $N(\mathcal{C})$ is the set of nodes, and $A(D, k)$ is the set of edges in the document. The sets $N(\mathcal{C})$ and $A(D, k)$ are defined as follows:

- The set $N(\mathcal{C})$ contains one node for each distinct word in the entire document corpus.
- The set $A(D, k)$ contains an **undirected edge** between nodes $i$ and $j$ if the word $i$ and word $j$ occur within a distance of **at most** $k$ positions. For example, for successive words, the value of $k$ is 1. The frequency of the edge is the number of times that word $i$ and word $j$ are separated by at most $k$ positions in the document.

The set $A(D, k)$ contains an **undirected** edge from each node to itself. The frequency of the edge is equal to the total number of times that the word occurs with distance $k$ of itself **in any direction**. Therefore, the frequency of the edge is **at least** equal to the frequency of the corresponding word in the document.

[2] The distance of a word to itself is zero.

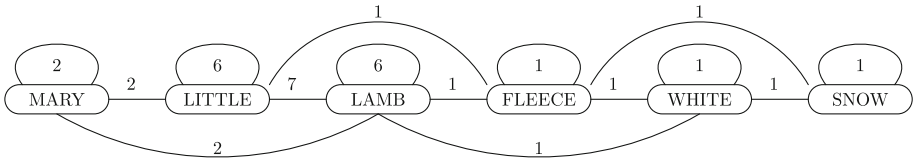


**Fig. 2** Illustration of distance graph representation (undirected graph of order 2)

In this first paper on distance graphs, we will not explore the undirected variation too extensively, but briefly mention it as an effective possibility for mining purposes. An illustration of the undirected distance graph for the example discussed earlier in this paper is provided in Fig. 2. In this case, we have illustrated the distance graph of order two. It is clear that the undirected distance graph can be derived from the directed graph by replacing the directed edges with undirected edges of the same frequency. In case edges in both directions exist, we can derive the frequencies of the corresponding undirected edge by adding the frequencies of the bi-directional edges. For example, the frequency of the undirected edge between "little" and "lamb" is the sum of the frequency of the directed edges in Fig. 1. The undirected representation loses some information about ordering, but still retains information on distances. While this paper is not focussed on this representation, we mention it, since it may be useful in many scenarios:

– Undirected graphs often provide a wider array of mining techniques, because undirected graphs are easier to process than directed graphs. This may be a practical advantage in many scenarios.
– While this paper is not focussed on cross-language retrieval, it is likely that directed graphs may be too stringent for such scenarios. While different languages may express the same word translations for a given text fragment, the ordering may be slightly different, depending upon the language. In such cases, the undirected representation may provide the flexibility needed for effective processing.

In future work, we will explore the benefits of the undirected variant of this problem. In the next section, we will discuss the applications of the distance graph representation.

## 3 Leveraging the distance graph representation: applications

One advantage of the distance graph representation is that it can be *used directly in conjunction with either existing text applications or with structural and graph mining techniques*, as follows:

– **Use with existing text applications:** Most of the currently existing text applications use the vector-space model for text representation and processing. It turns out that the distance graph *can also be converted to a vector-space representation*. The main property which can be leveraged to this effect is that the distance graph is sparse and the number of edges in it is relatively small compared to the total number of possibilities. For each edge in the distance graph, we can create a unique "token" or "pseudo-word". The frequency of this token is equal to the frequency of the corresponding edge. Thus, the new vector-space representation contains tokens only corresponding to such pseudo-words (including self-loops). *All existing text applications can be used directly in conjunction with this "edge-augmented" vector-space representation.*

– **Use with structural mining and management algorithms:** The database literature has seen an explosion of techniques [5,24,26–31,33] in recent years which exploit the underlying structural representation in order to provide more effective mining and management methods for text. Such approaches can sometimes be useful, because it is often possible to tailor the structural representation which is determined with this approach.

Both of the above methods have different advantages and work well in different cases. The former provides *ease in interoperability with existing text algorithms*, whereas the latter representation provides *ease in interoperability with recently developed structural mining methods*. We further note that while the vector-space representations of the distance graphs are larger than those of the raw text, the actual number of tokens in a document is typically only 4–5 times larger than the original representation. While this slows down the text processing algorithms, the slowdown is not large enough to become an unsurmountable bottleneck with modern computers. In the following, we will discuss some common text mining methods and the implications of the use of the distance graph representation with such scenarios.

### 3.1 Clustering algorithms

The most well-known and effective methods for clustering text [2,4,10,21,25,32] are variations on seed-based iterative or agglomerative clustering. The broad idea is to start off with a group of seeds and use iterative refinement in order to generate clusters from the underlying data. For example, the technique in [21] uses a variation of $k$-means clustering algorithms in which documents are assigned to seeds in each iteration. These assigned documents are aggregated and the low frequency words are projected out in order to generate the seeds for the next iteration. This process is repeated in each iteration, until the assignment of documents to seeds is stabilized. We can use *exactly the same algorithm directly on the vector-space representation of the distance graph*. In such a case, no additional algorithmic redesign is needed. We simply use the same algorithm, except that the frequency of the edges in the graph are used as a substitute for the frequencies of the words in the original document. Furthermore, other algorithms such as the EM clustering algorithm can be adapted to the distance graph representation. In such a case, we use the edges of the distance graph (rather than the individual words) in order to perform the iterative probabilistic procedures. In a sense, the edges of the graph can be considered *pseudo-words*, and the EM procedure can be applied with no changes. A second approach is to use the structural representation directly and determine the clusters by mining the frequent patterns in the collection and use them to create partitions of the underlying documents [5]. For example, consider the case, where a particular fragment of text "earthquake occurred in japan" or "earthquake in japan" occurs very often in the text. In such a case, this would result in the frequent occurrence of particular distance subgraph containing the words "earthquake", "occurred" and "japan". This resulting frequent subgraphs would be mined. This would tend to mine clusters which contain similar fragments of text embedded inside them.

### 3.2 Classification algorithms

As in the case of clustering algorithms, the distance graph representation can also be used in conjunction with classification algorithms. We can use the vector-space representation of the distance graphs directly in conjunction with most of the known text classifiers. Some examples are below:

– **Naive Bayes Classifier:** In this case, instead of using the original words in the document for classification purposes, we use the newly derived tokens, which correspond to the

edges in the distance graph. The probability distribution of these edges is used in order to construct the Bayes expression for classification. Since the approach is a direct analog of the word-based probabilistic computation, it is possible to use the vector-space representation of the edges directly for classification purposes.
- *k*-**Nearest Neighbor and Centroid Classifiers:** Since analogous similarity functions can be defined for the vector-space representation, they can be used in order to define corresponding classifiers. In this case, the set of tokens in the antecedent correspond to the edges in the distance graphs.
- **Rule Based Classifiers:** As in the previous case, we can use the newly defined tokens in the document in order to construct the corresponding rules. Thus, the left-hand side of the rules corresponds to combinations of edges, whereas the right hand side of the rules corresponds to class labels.

We can also leverage algorithms for structural classification [30]. Such algorithms have the advantage that they directly use the underlying structural information in order to perform the mining. Thus, the use of the distance graph representation allows for the use of a wider array of methods, as compared to the original vector-space representation. This provides us with greater flexibility in the mining process.

### 3.3 Indexing and retrieval

The structural data representation can also be used in conjunction with indexing and retrieval with two distinct approaches.

- We can construct an inverted representation directly on the augmented vector-space representation. While such an approach may be effective for indexing on relatively small sentence fragments, it is not quite as effective for document-to-document similarity search.
- We can construct structural indices directly on the underlying document collections [24,26–29,31,33] and use them for retrieval. We will see that the use of such an approach results in much more effective retrieval. By using this approach, it is possible to retrieve similar documents in terms of *entire structural fragments*. This is not possible with the use of the vector-space representation.

We note that the second representation *also allows us efficient document-to-document similarity search*. The inverted representation is only useful for search-engine like queries over a few words. Efficient document-to-document similarity indexing is an open problem for the case of text data (even with unaugmented vector-space representations). This is essentially because text data are inherently high dimensional, which is a challenging scenario for the similarity search application. *On the other hand, the structural representation provides off-the-shelf indexing methods, which are not available with the vector-space representation.* Thus, this representation not only provides more effective retrieval capabilities, but it also provides a wider array of such techniques.

An important observation is that *large connected subgraphs* which are shared by two documents typically correspond to text fragments shared by the two. Therefore, one can detect the nature and extent of structural similarity of documents to one another by computing the size of the largest connected components which are common between the original document and the target. This is equivalent to the problem of finding the maximum common subgraph between two data sets. We will discuss more on this issue slightly later.

### 3.4 Frequent subgraph mining on distance graphs: an application

Recently developed algorithms for frequent subgraph mining [26,29] can also be applied to large collections of distance graphs. Large connected subgraphs in the collection correspond to frequently occurring text fragments in the corpus. Such text fragments may correspond to significant textual characteristics in the collection. We note that such frequent pattern mining can also be performed directly on the vector-space model, though this can be inefficient since it may find a large number of disconnected graphs. On the other hand, subgraph mining algorithms [26,29] can be used to prune off many of the disconnected subgraphs from the collection. Therefore, the structural representation has a clear advantage from the perspective of discovering significant textual patterns in the underlying graphs. We can use these algorithms in order to determine the most frequently occurring text fragments in the collection. While the text fragments may not be exactly re-constructed from the graphs because of non-distinctness of the word occurrences, the overall structure can still be inferred from lower-order distance graphs such as $G_1$ and $G_2$. At the same time, such lower-order distance graphs continue to be efficient and practical for processing purposes.

#### *3.4.1 Plagiarism detection*

The problem of plagiarism detection *from large text collections* has always been very challenging for the text mining community because of the difficulty in determination of structural patterns from large text collections. However, the transformation of a text document to a distance graph provides a way to leverage techniques for graph pattern mining. We note that large connected graphs typically correspond to plagiarisms, since they correspond to huge structural similarities in the underlying text fragments for the document. In particular, the maximum common subgraph between a pair of graphs can be used in order to define the plagiarism index between two documents. Let $G^A$ and $G^B$ be the distance graph representation of two documents, and let $MCG(G^A, G^B)$ be the maximum common subgraph between the two documents. Then, we define the plagiarism index $\mathcal{P}(G^A, G^B)$ as follows:

$$\mathcal{P}\left(G^A, G^B\right) = \frac{\left|MCG\left\{G^A, G^B\right\}\right|}{\sqrt{|G^A|} \cdot \sqrt{|G^B|}} \tag{1}$$

We note that the computation of the maximum common subgraph is an NP-hard problem in the general case, but it is a simple problem in this case because of the fact that all node labels are distinct and correspond to unique words. Therefore, this approach provides an efficient methodology for detecting the likely plagiarisms in the underlying data.

## 4 Experimental results

Our aim in this section is to illustrate the *representational advantages* of the use of the distance graphs. This will be achieved with the use of *off-the-shelf vector-space and structural applications*. The aim is to minimize the specific effects of particular algorithms and show that the new representation does provide a more powerful expression of the text than the traditional vector-space representation. We note that further optimization of many of the (structural) algorithms is possible, though we leave this issue to future research. We will use diverse applications such as clustering, classification, and similarity search to show that this new representation does provide more qualitatively effective results. Furthermore, we will

use different kinds of off-the-shelf applications to show that our results are not restricted to a specific technique, but can achieve effective results over a wide variety of applications. We will also show that our methods maintain a level of efficiency which is only modestly less than the vector-space representation. This is a reasonable tradeoff in order to achieve the goals which are desired in this paper.

All our experiments were performed on an Intel PC with a 2.4 GHz CPU, 2GB memory, and running Redhat Linux. All algorithms were implemented and compiled by gcc 3.2.3.

### 4.1 Data sets

We choose three popular data sets used in traditional text mining and information retrieval applications in our experimental studies: (1) 20 newsgroups, (2) Reuters-21578, and (3) WebKB. Furthermore, the Reuters-21578 data set is of two kinds: Reuters-21578 R8 and Reuters-21578 R52. So we have four different data sets in total. The 20 newsgroups data set[3] contains 20,000 messages from 20 Usenet newsgroups, each of which has 1,000 Usenet articles. Each newsgroup is stored in a directory, which can be regarded as a class label, and each news article is stored as a separate file. The Reuters-21578 corpus[4] is a widely used test collection for text mining research. The data were originally collected and labeled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system[16]. Due to the fact that the class distribution for the corpus is very skewed, two sub-collections: Reuters-21578 R52 and Reuters-21578 R8, are usually considered for text mining tasks [11]. In our experimental studies, we make use of both of these two data sets to evaluate a series of different data mining algorithms. The WebKB data set [5] contains WWW-pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base project of the CMU text learning group. The 8,282 pages were manually classified into the following seven categories: student, faculty, staff, department, course, project, and other. Every document in the aforementioned data sets is preprocessed by eliminating non-alphanumeric symbols, specialized headers or tags and stop-words. The remaining words of each document are further stemmed by the Porter stemming algorithm. [6] The distance graphs are defined with respect to this post-processed representation.

Next, we will detail our experimental evaluation over a variety of data mining applications, including text classification, clustering, and similarity search. In many cases, we test more than one different method. The aim is to show that the distance graph has a number of representational advantages for mining purposes over a wide variety of problems and methods.

### 4.2 Classification applications

In this section, we will first test the effectiveness of the distance graph representation on a variety of classification algorithms. We make use of Rainbow [19], a freely available statistical text classification toolkit for our experimental studies. First of all, Rainbow reads and indexes text documents and builds the statistical model. Then, different text classification algorithms are performed upon the statistical model. We used three different algorithms from Rainbow for text classification. These algorithms are the Naive Bayes classifier [18], TFIDF

[3] http://kdd.ics.uci.edu/databases/20newsgroups.

[4] http://kdd.ics.uci.edu/databases/reuters21578.

[5] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz.

[6] http://tartarus.org/martin/PorterStemmer.

classifier [15], and Probabilistic Indexing classifier [12], respectively. For each classification method of interest, we employ the vector-space models including unigram, bigram, and trigram models, and the distance graph models of different orders ranging from 1 to 4, respectively, as the underlying representational models for text classification. In order to simulate the behaviors of the bigram model and the trigram model, we extract the most frequent 100 doublets and triplets from the corpora and augment each document with such doublets and triplets, respectively. The vector-space models are therefore further categorized as unigram with no extra words augmentation, bigram with doublets augmentation and trigram with triplets augmentation. We conduct 5-fold cross-validation for each algorithm in order to compare the classification accuracies derived from different representation strategies. All the reported classification accuracies are statistically significant with 95 % significance level.

In Fig. 3, we have illustrated the classification accuracy results in the 20 newsgroups data set for the three different classifiers. In addition to the vector-space representations for unigram, bigram, and trigram models, we have also illustrated the classification results for the distance graph representations with different distance orders ranging from 1 to 4. It is clear that the addition of structural information in the distance graph models improves the quality of the underlying result in most cases. Specifically, the best classification results are obtained for distance graphs of order 2 in Naive Bayes classifier, and of order 1 in TFIDF classifier and of order 4 in Probabilistic Indexing classifier, respectively. Meanwhile, in all of the cases, the distance graph representations consistently obtain better classification results than all the vector-space models, including the unigram, the bigram, and the trigram models. Even though the optimal classification accuracy is achieved for distance graphs of orders 1 and 2 in some experimental scenarios, it is noteworthy that the vector-space representations did not even perform better than the higher order distance graphs in all cases. We also tested the classification results for the Reuters-21578 (R8 and R52) data sets. The classification accuracy results are illustrated in Figs. 4 and 5, respectively. It is evident that the distance graph representations are able to provide a higher classification accuracy over the different kinds of classifiers as compared to the vector-space representations. The reason for this is that the distance graph representations can capture structural information about the documents which is used in order to help improve the classification accuracy. As a result, the classification results obtained with the use of the distance graph representations are superior to those obtained using the vector-space representations.

We also tested the efficiency of the distance graph representations for the different data sets. We note that the higher order distance graph representations have a larger number of edges and are therefore likely to be somewhat slower. The results for the 20 newsgroups

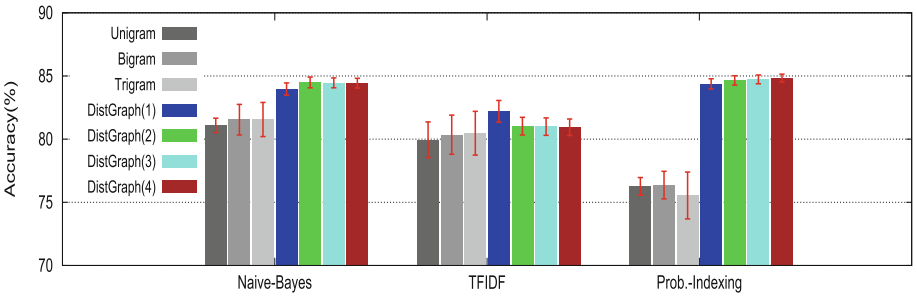


**Fig. 3** Text classification accuracy with 95 % confidence level (20 newsgroups)

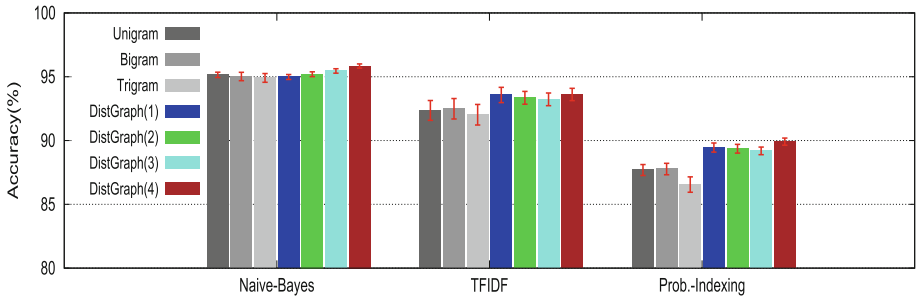


**Fig. 4** Text classification accuracy with 95 % confidence level (Reuters-21578 R8)

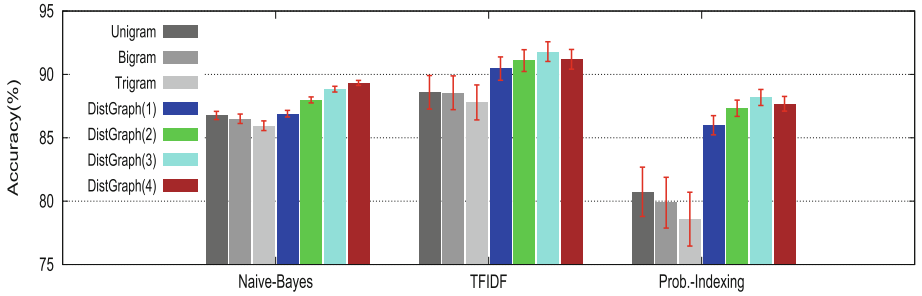


**Fig. 5** Text classification accuracy with 95 % confidence level (Reuters-21578 R52)

data set, Reuters-21578 R8, and Reuters-21578 R52 data sets are illustrated in Fig. 6a–c respectively. In each case, the running times are illustrated on the $Y$-axis, whereas the order of the distance graph representation is illustrated on the $X$-axis. In this graph, the order-0 representation corresponds to the vector-space representation (unigram). It is evident that the running time increases only linearly with the order of the representation. Since the optimal results are obtained for lower-order representations, it follows that only a modest increase in running time is required in order to improve the quality of the results with the distance graph representations.

### 4.3 Clustering application

We further tested our distance graph representation for the clustering application. Two different text clustering algorithms are adopted in our experimental studies: K-means [14] and Hierarchical EM clustering [7]. We implemented the K-means algorithm and used the Hierarchical EM clustering algorithm in crossbow, which provides clustering functionality in the Rainbow toolkit [19]. For each clustering algorithm, we used *entropy* as a measure of quality of the clusters [23] and compare the final entropies of clusters generated with different underlying representations. The entropy of clusters can be formally defined as follows: Let $C$ be a clustering solution generated by a specific clustering algorithm mentioned above. For each cluster $c_j$, the class distribution of the data within $c_j$ is computed first: we denote $p_{ij}$ as the probability that a member of $c_j$ belongs to class $i$. Then, the entropy of $c_j$ is computed as follows:

$$E_{c_j} = -\sum_i p_{ij} \log(p_{ij}) \tag{2}$$

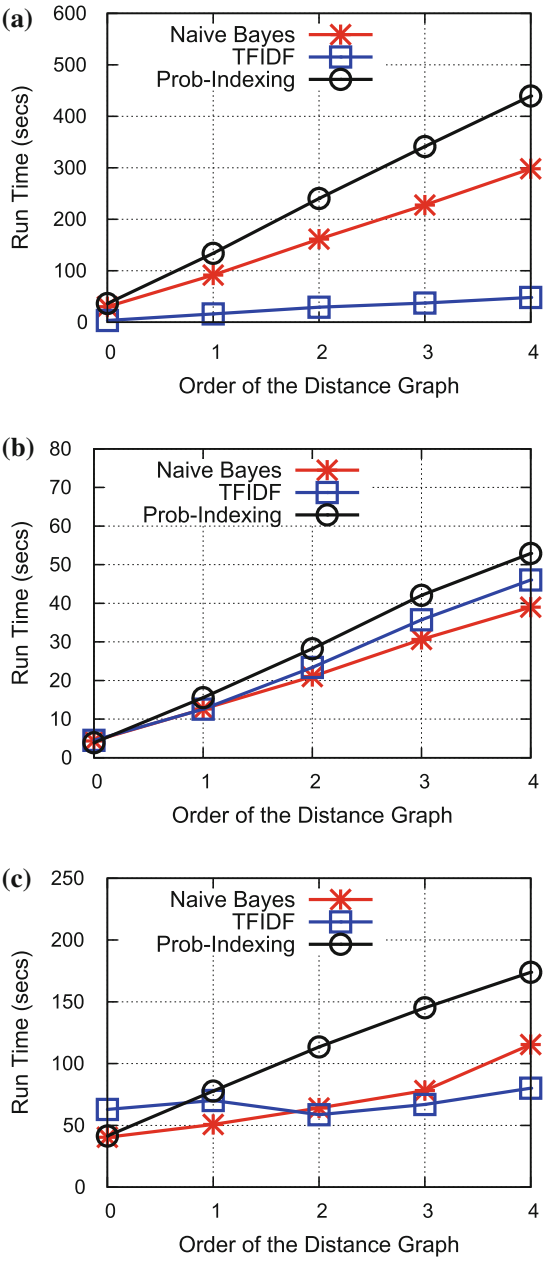


**Fig. 6** Classification efficiency on different data sets. **a** 20 Newsgroups, **b** Reuters 21578 R8, **c** Reuters 21578 R52

where the sum is taken over all classes. The total entropy for a set $m$ of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, as follows:

$$E_C = \sum_{j=1}^{m} \frac{|c_j| \times E_{c_j}}{\sum_{j=1}^{m} |c_j|} \tag{3}$$

where $|c_j|$ is the size of cluster $c_j$.

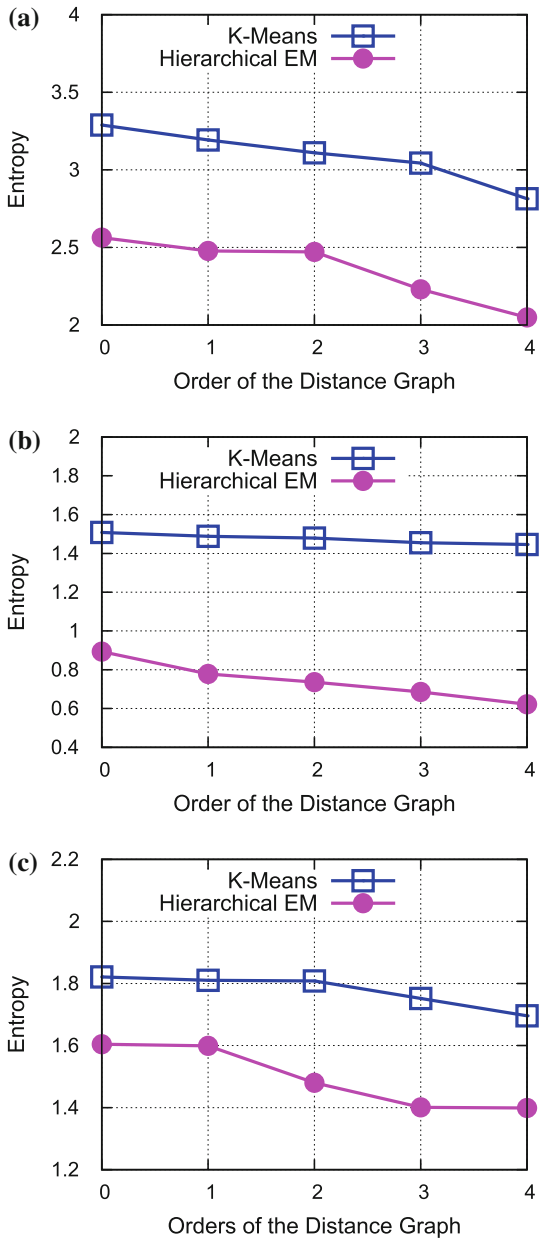


**Fig. 7** Entropy of clustering results on different data sets. **a** 20 Newsgroup, **b** Reuters 21578 R8, **c** WebKB

We tested both the K-means and the Hierarchical EM clustering algorithm with the use of the vector-space representation as well as the distance graph method. The results are illustrated in Fig. 7 for the 20 newsgroups data set (Fig. 7a), the Reuters-21578 R8 data set (Fig. 7b), and the WebKB data set (Fig. 7c), respectively. The order of the distance graph is illustrated on the $X$-axis, whereas the entropy is illustrated on the $Y$-axis. The standard vector-space representation corresponds to the case when we use a distance graph of order-0

(unigram). It is evident from the results of Fig. 7, that the entropy reduces with increasing order of the distance graph. This is because the distance graph uses the structural behavior of the underlying data in order to perform the distance computations. The higher quality of these distance computations also improves the corresponding result for the overall clustering process.

We also tested the efficiency of different clustering methods with increasing order of the distance graph representation on different data sets. The results for the 20 newsgroups, the Reuters-21758 R8, and the WebKB data sets are illustrated in Fig. 8a–c, respectively. The order of the distance graph is illustrated on the $X$-axis, whereas the running time is illustrated on the $Y$-axis. It is clear that the running time increases gradually with the order of the distance graph. The linear increasing in running time is an acceptable tradeoff, because of the higher quality of the results obtained with the use of the method. Furthermore, since the most effective results are obtained with the lower-order distance graphs, this suggests that the use of the method provides a significant advantage without significantly compromising efficiency.

### 4.4 Similarity search application

We also tested the effectiveness of our distance graph presentation in the similarity search application. For the case of the distance graph representation, we used the similarity measure, which uses the cosine on the edge-based structural similarity as defined by a frequency-weighted version of Eq. 1. We compared the effectiveness of our approach to that of the (standard) vector-space representations, including unigram, bigram and trigram. Similar to the text classification case, we augment each of the documents with most frequent 100 doublets and triplets extracted from the text corpora to simulate the behaviors of the bigram model and the trigram model, respectively. A key issue of similarity search is the choice of the metric used for comparing the quality of search results with the use of different representations. In order to measure the qualitative performance, we used a technique which we refer to as the *class stripping* technique. We stripped off the class variables from the data set and found the $k = 30$ nearest neighbors to each of the records in the data set using different similarity methods. In each case, we computed the number of records for which the majority class matched with the class variable of the target document. If a similarity method is poor in discriminatory power, then it is likely to match unrelated records and the class variable matching is also likely to be poor. Therefore, we used the class variable matching as a surrogate for the effectiveness of our technique. The results for the WebKB data set and the Reuters-21578 R8 data set are illustrated in Table 1. It is evident that in most cases, the quality of the similarity search is better for the higher order distance graphs. The results for these lower-order representations were fairly robust and provided clear advantages over the vector-space representations for all unigram, bigram, and trigram. Thus, the results of this paper suggest that it is possible to improve the quality and effectiveness of text processing algorithms with the use of novel distance graph models.

## 5 Conclusions and summary

In this paper, we introduced the concept of distance graphs, a new paradigm for text representation and processing. The distance graph representation maintains information about the relative placement of words with respect to each other, and this provides a richer representation

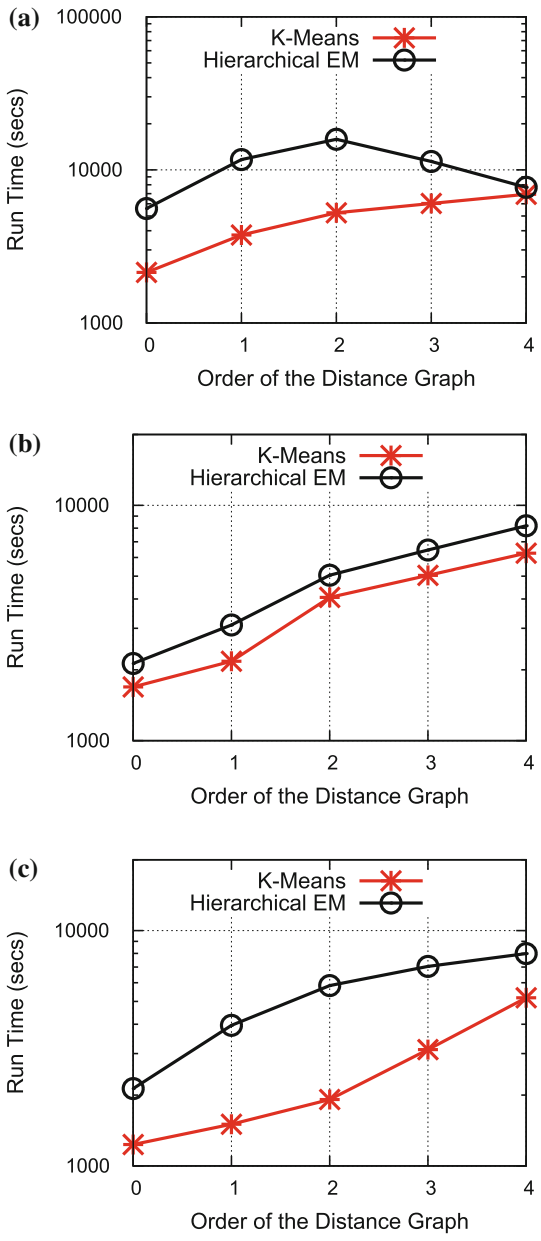


**Fig. 8** Clustering efficiency on different data sets. **a** 20 Newsgroups, **b** Reuters 21578 R8, **c** WebKB

for mining purposes. We can use this representation in order to exploit the recent advancements in structural mining algorithms. Furthermore, the representation can be used with minimal changes to existing data mining algorithms if desired. Thus, the new representation does not require additional development of new data mining algorithms. This is an enormous advantage, since existing text processing and graph mining infrastructure can be used directly with the distance graph representation. In this paper, we tested our approach

**Table 1** Similarity search effectiveness

| Representation | WebKb | Reuters-21758 R8 |
|---|---|---|
| Vector space (unigram) | 44.99 | 82.80 |
| Vector space (bigram) | 45.15 | 82.93 |
| Vector space (trigram) | 45.19 | 82.69 |
| DistGraph(1) | 45.91 | 83.50 |
| DistGraph(2) | 45.55 | 83.34 |
| DistGraph(3) | 45.62 | 83.31 |
| DistGraph(4) | 48.11 | 81.0 |

with a large number of different classification, clustering, and similarity search applications. Our results suggest that the use of the distance graph representation provides significant advantages from an effectiveness perspective.

In future work, we will explore specific applications, which are built on top of the distance graph representation in greater detail. Specifically, we will study the problems of similarity search, plagiarism detection, and its applications. We have already performed some initial work on performing similarity search, when the target is a set of documents [6], rather than a single document. We will also study how text can be efficiently indexed and retrieved with the use of the distance graph representation.

## References


1. Aggarwal C (2010) Managing and mining graph data. Springer, Berlin
2. Aggarwal C, Yu PS (2011) On clustering massive text and categorical data streams. Knowl Inf Syst 24(2):171–196
3. Aggarwal C, Zhao P (2010) Graphical models for text: a new paradigm for text representation and processing. In: The 33rd international ACM SIGIR conference on research and development in, information retrieval (SIGIR'10), pp 899–900
4. Aggarwal C, Zhai C (2012) A survey of text clustering algorithms, Mining text data. Springer, Berlin
5. Aggarwal C, Ta N, Feng J, Wang J, Zaki MJ (2007) XProj: a framework for projected structural clustering of XML documents. In: The 13th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'07), pp 46–55
6. Aggarwal C, Lin W, Yu PS (2012) Searching by corpus with fingerprints. In: EDBT conference, pp 348–359
7. Bishop C (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
8. Cavnar W, Trenkle J (1994) N-gram based text categorization. In: Symposium on document analysis and, information retrieval (SDAIR'94), pp 161–194
9. Collins MJ (1996) A new statistical parser based on bigram statistical dependencies. In: The 34th annual meeting on association for, computational linguistics (ACL'96), pp 184–191
10. Cutting D, Karger D, Pedersen J, Tukey J (1992) Scatter/gather: a cluster-based approach to browsing large document collections. In: The 15th annual international ACM SIGIR conference on research and development in, information retrieval (SIGIR'92), pp 318–329
11. Debole F, Sebastiani F (2004) An analysis of the relative hardness of reuters-21578 subsets. J Am Soc Inf Sci Tech 56(6):584–596
12. Fuhr N, Buckley C (1990) Probabilistic document indexing from relevance feedback data. In: The 13th annual international ACM SIGIR conference on research and development in, information retrieval (SIGIR'90), pp 45–61
13. Hearst M (1992) Automatic acquisition of hyponyms from large text corpora. In: The 14th conference on, computational linguistics (COLING'92), pp 539–545
14. Jain A, Dubes R (1988) Algorithms for clustering data. Prentice-Hall, New Jersey
15. Joachims T (1997) A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: The fourteenth international conference on, machine learning (ICML'97), pp 143–151

16. Lewis DD (1996) reuters-21578 data set. http://www.daviddlewis.com/resources/test-collections/reuters21578
17. Lin D (1998) Extracting collocations from text corpora. In; First workshop on computational terminology
18. Maron M (1961) Automatic indexing: an experimental inquiry. J ACM 8(3):404–417
19. McCallum A (1996) Bow: a toolkit for statistical language modeling. Text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow
20. Salton G, Mcgill MJ (1983) An introduction to modern information retrieval. McGraw Hill, New York
21. Schutze H, Silverstein C (1997) Projections for efficient document clustering, In: The 20th annual international ACM SIGIR conference on research and development in, information retrieval (SIGIR'97), pp 74–81
22. Smadja F (1993) Retrieving collocations from text: Xtract. Comput Linguist 19(1):143–177
23. Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: KDD text mining, workshop, pp 109–111
24. Wang H, Park S, Fan W, Yu PS (2003) ViST: a dynamic index method for querying XML data by tree structures. In: The 2003 ACM SIGMOD international conference on management of data (SIGMOD'03), pp 110–121
25. Wang F, Zhang C, Li T (2007) Regularized clustering for documents. In: The 30th annual international ACM SIGIR conference on research and development in, information retrieval (SIGIR'07), pp 95–102
26. Yan X, Han J (2003) CloseGraph: mining closed frequent graph patterns. In: The ninth ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), pp 286–295
27. Yan X, Yu PS, Han J (2005a) Graph indexing based on discriminative frequent structure analysis. ACM Trans Database Syst 30(4):960–993
28. Yan X, Yu PS, Han J (2005b) Substructure similarity search in graph databases. In: The 2005 ACM SIGMOD international conference on management of data (SIGMOD'05), pp 766–777
29. Yan X, Cheng H, Han J, Yu PS (2008) Mining significant graph patterns by scalable leap search. In: The 2008 ACM SIGMOD international conference on management of data (SIGMOD'08), pp 433–444
30. Zaki MJ, Aggarwal C (2003) XRules: an effective structural classifier for XML data. In: The ninth ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), pp 316–325
31. Zhao P, Han J (2010) On graph query optimization in large networks. PVLDB 3(1):340–351
32. Zhao Y, Karypis G (2004) Empirical and theoretical comparisons of selected criterion functions for document clustering. Mach Learn 55(3):311–331
33. Zhao P, Xu J, Yu PS (2007) Graph indexing: tree + delta >= graph. In: The 33rd international conference on very large data bases (VLDB'07), pp 938–949

## Author Biographies

**Charu C. Aggarwal** is a Research Scientist at the IBM T. J. Watson Research Center in Yorktown Heights, New York. He completed his B.S. from IIT Kanpur in 1993 and his Ph.D. from Massachusetts Institute of Technology in 1996. His research interest during his Ph.D. years was in combinatorial optimization (network flow algorithms), and his thesis advisor was Professor James B. Orlin . He has since worked in the field of performance analysis, databases, and data mining. He has published over 200 papers in refereed conferences and journals, and has filed for, or been granted over 80 patents. Because of the commercial value of the above-mentioned patents, he has received several invention achievement awards and has thrice been designated a Master Inventor at IBM. He is a recipient of an IBM Corporate Award (2003) for his work on bio-terrorist threat detection in data streams, a recipient of the IBM Outstanding Innovation Award (2008) for his scientific contributions to privacy technology, and a recipient of an IBM Research Division Award (2008) and IBM Outstanding Technical Achievement Award (2009) for his scientific contributions to data stream research. He has served on the program committees of most major database/data mining conferences, and served as program vice-chairs of the SIAM Conference on Data Mining, 2007, the IEEE ICDM Conference, 2007, the WWW Conference 2009, and the IEEE ICDM Conference, 2009. He served as an associate editor of the IEEE Transactions on Knowledge and Data Engineering Journal from 2004 to 2008. He is an associate editor of the ACM TKDD Journal, an action editor of the Data Mining and Knowledge Discovery Journal, an associate editor of the ACM

SIGKDD Explorations, and an associate editor of the Knowledge and Information Systems Journal. He is a fellow of the IEEE for "contributions to knowledge discovery and data mining techniques", and a life-member of the ACM.

**Peixiang Zhao** is an Assistant Professor of the Department of Computer Science at the Florida State University at Tallahassee, Florida. He obtained his Ph.D. in Computer Science in 2012 from the University of Illinois at Urbana-Champaign under the supervision of Professor Jiawei Han. He also obtained a Ph.D. in 2007 at the Department of Systems Engineering and Engineering Management from the Chinese University of Hong Kong, under the supervision of Professor Jeffrey Xu Yu. Peixiang obtained his B.S. and M.S. degree from Department of Computer Science and Technology (it now becomes School of Electronics Engineering and Computer Science) in Peking University at 2001 and 2004, respectively. Peixiang's research interests lie in database systems, data mining, and data-intensive computation and analytics. More specifically, he has been focused on the problems in modeling, querying and mining graph-structured data, such as large-scale graph databases and information networks, which have numerous applications in bioinformatics, computer systems, business processes, and the Web. Peixiang is a member of the ACM SIGMOD.