Graphical Models for Text: A New Paradigm for Text Representation and Processing

Charu C. Aggarwal
IBM T. J. Watson Research Center
Hawthorne, New York, USA
charu@us.ibm.com

Peixiang Zhao
University of Illinois at Urbana-Champaign
Urbana, Illinois, USA
pzhao4@uiuc.edu

ABSTRACT

Almost all text 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 distance and ordering information between the words, and provide a much richer representation of the underlying text. 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.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval Models

General Terms: Algorithms

1. INTRODUCTION

The most common representation for text is the *vector-space representation*. 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. For many applications, such an approach can lose key analytical insights. This is especially the case for applications in which the structure of the document plays a key role in the underlying semantics. 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 straitjacket on the development of richer representations of text, this constraint has become easier to overcome in recent years. This is because of advances in the computational power of hardware, and the increased sophistication of algorithms in other fields such as graph mining, which can be leveraged with representations such as those discussed in this paper. 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 show that such a representation can retain much richer information about the underlying data. It also allows for the use of many current *text and graph mining algorithms without the need for new algorithmic efforts for this new representation*. 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. In some cases, it also enables distance-based applications which are not possible with the vector-space representations.

2. DISTANCE GRAPHS

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 complete 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 make it an effective representation for easy 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 1. *A distance graph of order k for a document D in 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:*

(a) *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 $\mathcal{C}$.*

(b) *The set $A(D, k)$ contains a directed edge from node i to node j*

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
ACM 978-1-60558-896-4/10/07.

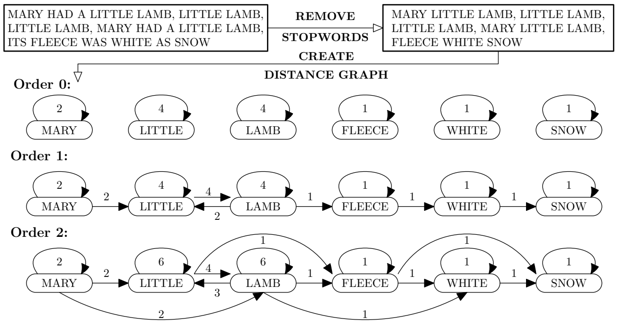

Figure 1: Illustration of Distance Graph Representation

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 default, 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 (known as *stop-words*), which are typically filtered out before text processing. Therefore, 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 higher orders provide structural insights of different levels of complexity. An example of the distance graph representation for a well-known nursery rhyme *"Mary had a little lamb"* is illustrated in Figure 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 or order 0 contain only self loops. The frequencies of these self-loops in the order-0 representation corresponds 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 self-loop 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.

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.

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 results presented in [1], it suffices to use low values of k for effective processing in most mining applications.

PROPERTY 1. *Let $f(D)$ denote the number of words 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 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 semantic richness captured by the distance graph representation.

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:
(a) 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.*
(b) Use with structural mining and management algorithms: The database literature has seen an explosion of management and mining techniques for graph and XML mining. Since our distance-based representation can be naturally expressed as a graph or XML document, such techniques can also be used in conjunction with the distance graph representation. The advantages of such approaches are that they are specifically tailored to graph data, and can therefore determine novel insights in the underlying word-distances.

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*. In [1], we have presented details of methods and corresponding experimental results for problems such as clustering, classification, similarity search and plagiarism detection. We illustrate advantages both in terms of accuracy and the enablement of distance-based applications.

3. REFERENCES

[1] C. Aggarwal, P. Zhao. Graphical Models for Text: A New Paradigm for Text Representation and Processing, *IBM Research Report*, 2010.