International Journal of Smart Home
Vol.8, No.4 (2014), pp.37-46
http://dx.doi.org/10.14257/ijsh.2014.8.4.04

Research Trends on Graph-Based Text Mining

Jae-Young Chang and Il-Min Kim

Dept. of Computer Engineering, Hansung University
2-Ga SamSun-Dong, SungBuk-Gu, Seoul, 136-792, Korea
{jychang, ikim}@hansung.ac.kr

Abstract

Since text mining has been assumed to apply for unformatted text (document), it is necessary to represent text with simplified models. One of the most commonly used models is the vector space model, in which text is represented as a bag of words. Recently, many researches tried to apply a graph-based text model for representing semantic relationships between words. In this paper, we surveyed research trends of graph-based text representation models for text mining. We summarized the models, their features and forecasted further researches.

Keywords: *Text Mining, Vector Space Model, Text Representation, Graph Model*

1. Introduction

Text mining is an area of data mining and its goal is to analyze unformatted text and to find out the hidden knowledge of the text. Traditionally, one of the most commonly used models for representing text is VSM (Vector Space Model) [1], in which frequently used words and their weights were expressed as vectors. However, because of the simplicity of VSM, we could not fully express semantic meaning and context with VSM. In order to solve the problems of the VSM, text mining researches based on graph model have been actively carried out after year 2000. An advantage of text representing model based on graph is that you can clarify the document meaning with words, phrases, concepts, and relations.

In this paper, we will analyze the research trends of text mining which are based on graph. At first, we will delve into researches on the vector space models, and then, we will systematically summarize the graph-based text models according to their characteristics.

2. Vector Space Model

In VSM, a text is represented as a point in an n-dimensional space. Since VSM can represent an atypical text with a simple and formulaic notation, various algorithms, which had been used in data mining, can be applied without any modification. Because of the advantage, many researches on VSM are being actively carried out. However, VSM also has following disadvantages due to its simplicity [2].

- If two documents use different words for similar meaning, their similarities cannot be computed easily.
- The meaning of a text or the structure of a text cannot be expressed in VSM.
- Word appearance sequence or word relationship cannot be represented in VSM.

ISSN: 1975-4094 IJSH
Copyright ⓒ 2014 SERSC

Various researches have been carried out in order to solve these problems described above. Until the present time, the graph-based text representation model has been recognized as one of best solutions for these problems.

3. Classification of Graph-based Text Model

3.1. Classification by Graph Format

A graph *G* is derived from a text or a text set can be expressed as follows:

$G = \{ V, E \}$

In this notation, *V* and *E* represent the set of nodes and edges respectively. With this notation, we can represent various graphs models according to the definitions of *V* and *E*. In this paper, we classified the graphs in detail according to the node representation and the edge representation.

3.1.1. Node Representation Method: Nodes of graph *G* represent text components, such as, words, sentences, paragraphs, and texts themselves. Also, nodes can represent concepts which could be considered as semantic components. According to the definition of the model, a node can represent one component, or more than one component. If a node represents one component, it is called homogenous representation. If a node represents more than two components, it is called heterogeneous representation. In addition, nodes can be either weighted or unweighted, depending on whether or not assigning weighted value into the nodes.

Homogenous vs. Heterogeneous Representation

In common notations used in homogenous representations, a node indicates a word as shown in Figure 1 [3-8]. Co-occurrence information between two words is usually expressed in the graph. Co-occurrence means that two related words are appeared at the same time in a sentence. In that case you need to connect the words with an edge. Grammatical associations between words or semantic similarities could be expressed using a graph [10]. Since this representation is simple, the cost for building and analyzing a model is low. Another advantage of this representation is that the existing algorithms used in the vector space model can be applied without any modification. In some researches, they also used homogenous representations, in which sentences, paragraphs, or concepts are represented as nodes [12, 17, 18].

In heterogeneous representations, more than two different typed components which could be words, sentences, texts or concepts are represented as nodes. One of the most common heterogeneous representations is a bipartite graph [11, 15]. For example, Figure 2 shows an example of the bipartite graph composed of documents and concepts.

Copyright ⓒ 2014 SERSC

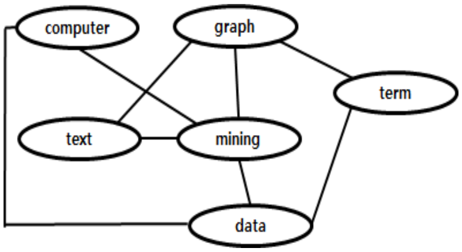

Figure 1. An Example of Homogeneous Representation Model

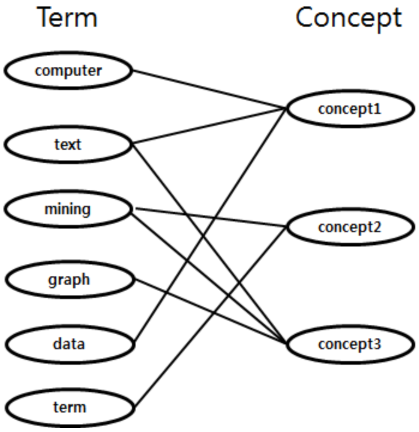

Figure 2. An Example of Bipartite Graph

Weighted and Unweighted

In weighted representations, a weighted value has been assigned in each node. In contrast, if a node does not have weighted value, it is an un-weighted representation. Most researchers assume that a weighted value indicate the importance of a node in the graph. In order to evaluate the weight value of a node, some researches, for example PageRank[19], calculated the weight value indirectly with considering the number of edges which are connected to the node, the weights of the edges, and the weights of the neighboring nodes.

3.1.2. Edge Representation Method: An edge represents the relationship between nodes. Edges can be classified by three characteristics. Edges can be either directed or undirected. Edges can be either weighted or un-weighted. Edges can be either labeled or unlabeled. Each characteristic will be explained in detail.

Directed vs. Undirected Edges

Directed edges are used for indicating the orders of nodes or the mutual interactions between nodes. For example, if you want to indicate the orders of words, you need to use directed edges in the graph [4]. Each word in a sentence has its grammatical role (for example, subject, verb, and object). Directed edges [9, 10] can be used for representing their grammatical roles. A tree could be used for depicting grammatical roles, since it could be considered a directed edge representation [13].

If there are no orders or mutual interaction between nodes, undirected edges are used for connecting the related nodes. If you need to represent the co-occurrences between words, you would use undirected edges in the graph [6, 18].

Copyright ⓒ 2014 SERSC

Weighed vs. Un-weighted Edges

The weights of edges are numbers which indicate the relationships of nodes. A weight could indicate the frequency that related words would appear at the same time in a co-occurrence graph [3, 4]. The weight of an edge could indicate the distance of two words in a text. If two nodes are not related quantitatively, un-weighted edges are commonly used [5, 8, 9, 10].

Labeled vs. Unlabeled Edges

Labeled edges were used in some graph models [5, 7, 9, 10, 13]. Labels represent the roles of the edges, in which labels indicate the relationships between words. In the paper [10], the edge is labeled 'verb' when the edge connects from the 'subject' node to the 'object' node. In the paper [9], a sentence is depicted as a parsing tree. The label depicts the PoS(part of speech) of each word. In order to indicate the grammatical role of each word, labeled edges are generally used. Except the cases described above, unlabeled edges are commonly used.

3.2. Classification by Graph Contents

Another classification is based on the graph contents. Graph contents can be classified into three models. The first of the three models represents the co-occurrence or the similarity between nodes. The second model represents grammatical relationship between nodes. The last model represents the semantic relationships between nodes.

3.2.1. Co-occurrence or Similarity Express Model: This model has been more commonly used in the previous researches than other models. This model represents the co-occurrence information between words or the similarity between sentences [3, 4, 6, 16, 17, 18]. This model is simpler than other models and costs less to build a graph. In addition, various algorithms which have been proposed in graph mining area can easily be applied to this model. Since this model is language independent, many algorithms have been applied to English text can also be applied to other language texts.

3.2.2. Grammatical Relationship Model: This model represents the grammatical relationship and also can represent the dependency among nodes with the labels of edges [9, 10]. This model has an advantage that sentence structures can be clearly specified. This model also has the disadvantage of the high computation cost because of the complexity of the graph.

3.2.3. Semantic Association Model: In this model, a node represents a concept. One example of this model depicted the relationships between text and concept with a bipartite graph [15]. Important words are considered as concepts and the relationships between texts and concepts are expressed as a bipartite graph. In one of the other examples, the first step is to select representative concepts. After selecting concepts, the next step is to build a tree which represents the relationship between the concepts. This model requires a concept database such as Wikipedia [11].

4. Technologies for Graph-based Text Mining

We will clearly describe major technologies and algorithms that have been used for graph-based text mining in this section. Most current researches on graph-based text mining have adopted existing algorithms and well-known technologies, which have been properly adjusted to the situations. In the graph-based representation models, you need to compute the weights of nodes or edges.

The simple method for computing the weight of a node is to compute the appearance frequency of the corresponding word or to compute TF-IDF. The weight of an edge can be calculated by the co-occurrence frequency of the connected nodes. If a node represents a sentence, the co-occurrence cannot be computed. Therefore, the methods for computing the sentence similarity can be applied here. The sentence similarity can be computed by using the cosine similarity or the Euclidean distance. Other major technologies will be delineated as follows.

4.1. PageRank

PageRank is one of well-known algorithms for graph ranking. In the beginning, PageRank was used for ranking web pages in World Wide Web. Later, PageRank has been widely used for ranking nodes in graphs. The weight of the node V_i is computed by the importance and the number of the connected nodes. $PR(V_i)$, the weight of V_i, is defined as follows.

$$PR(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{PR(V_j)}{\left|Out(V_j)\right|} \tag{1}$$

In the equation (1), $In(V_i)$ represents the node set of pointing to V_i and $Out(V_i)$ represents the node set of pointing out of V_i. The damping factor *d* can vary from 0 to 1 and the default value of *d* is 0.85 [19]. PageRank was used to calculate the node weights for summarizing and classifying documents [6, 16, 17, 23].

4.2. TextRank and LexRank

TextRank algorithm, a variation of PageRank, is a ranking algorithm and is based on graphs. TextRank algorithm considers undirected graphs by default. In order to calculate the similarity, TextRank needs to consider the weights of edges. For example, the weights of nodes could be computed as follows.

$$PR(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\left|\sum_{V_k \in Out(V_j)} w_{jk}\right|} PR(V_j) \tag{2}$$

In the equation (2), w_{ij} represents the weight of the edge connecting node V_i and V_j. Though TextRank algorithm originally was developed for summarizing document, it is also applied to novelty search [25].

LexRank algorithm, which was similar to TextRank, was also proposed. The two algorithms have different application fields. LexRank was developed for summarizing the large documents, and TextRank was proposed for multi-document summarization. Except the application field, the two algorithms are very similar.

4.3. HITS

HITS(Hyperlink-Induced Topic Search) algorithm was developed for ranking web pages. HITS was developed earlier than PageRank and had a big impact on developing the

Copyright ⓒ 2014 SERSC

PageRank algorithm [29]. In HITS, if a web page is linked with important pages, the rank of the page will be raised.

This algorithm can be applied for computing the weights of the nodes in a graph model [8, 16, 17, 26]. In HITS, authority point and hub point for every node are assigned. The authorities represent the number of incoming links and hubs represent the number of out-coming links. These points can be calculated as follows

$$HITS_A(V_i) = \sum_{V_j \in In(V_i)} HITS_H(V_j)$$
$$HITS_H(V_i) = \sum_{V_j \in Out(V_i)} HITS_A(V_j) \qquad (3)$$

The weight of each node can be computed by summing or averaging the authority points and hub points.

4.4. PMI

PMI(Pointwise Mutual Information) is one of methods that have been used for measuring the relationship of two objects. PMI is used for computing the weights of edges in graphs [15, 25]. Assume *P*(*i*) is the probability that word w_i occurs in the document of a co-occurrence graph and *P*(*i*, *j*) is the probability that words w_i and w_j occur at the same time. PMI for w_i and w_j can be computed as follows:

$$PMI_{ij} = \log_2 \frac{P(i, j)}{P(i)P(j)} \qquad (4)$$

If PMI value approaches 0, a word becomes independent from the other. If PMI value approaches 1, the two words are more closely related.

4.5. Frequent Itemset Mining

Frequent Itemset Mining is also called association rule mining, which has been used for searching co-occurrence item set. If two nodes are included in a frequent item set, this method is used for assigning the weight of the edge [6]. Frequent Itemset Mining can also be applied for searching a frequent sub-graph [22].

5. Conclusions and Further Research Trends

This paper explained and classified the previous text-representing models based on graphs. Table 1 summarized major researches on this area. The first column of Table 1 depicts the reference number, the second column is the application area, and the third and fourth column indicates the features of nodes and edges respectively. The fifth column is what the graph tried to express.

As we surveyed in this paper, graph-based text representing models adopted different modeling according to their goals and application areas. In other words, there were little effort to standardize the graph models. Therefore more systematic research on graph model for representing text is required. The systematic research for building standard graph models will be useful for document classification, aggregation, summary, search, and will be applied for various document analyses.

Table 1. Representative Researches of Graph-Based Text Mining

Ref. No.	Application Area	Graph Structure		Graph Contents
		Node	Edge	
3	classification	homo(term)	directed, weighted, unlabeled	co-occurrence
4	clustering	homo(term)	directed, weighted, unlabeled	co-occurrence
5	search	homo(term)	directed, unweighted, labeled	co-occurrence, syntax
6	classification	homo(term)	undirected, weighted, unlabeled	co-occurrence
7	classification	homo(term)	directed, unweighted, labeled/unlabeled	co-occurrence, syntax
8	summarization	homo(term)	directed, unweighted, unlabeled	co-occurrence
9	classification	hetero(term+PoS)	directed, unweighted, labeled	syntax, semantic
10	summarization	homo(term)	directed, unweighted, labeled	syntax
11	classification clustering	hetero(doc+concept)	directed/undirected, weighted, unlabeled	semantic (bipartite graph)
12	summarization	homo(sentence)	undirected, weighted, unlabeled	similarity
13	opinion mining	homo(term)	directed, weighted, labeled	semantic tree
14	summarization	homo(sentence)	undirected, weighted, unlabeled	similarity
15	clustering	hetero(doc+concept)	undirected, unweighted, unlabeled	semantic (bipartite graph)
16	summarization	homo(sentence)	directed/undirected, weighted/unweighted, unlabeled	similarity
17	summarization	homo(sentence)	directed/undirected, weighted, unlabeled	similarity
18	keyword extraction, summarization	homo (term or sentence)	directed/undirected, weighted/unweighted, unlabeled	co-occurrence, similarity
20	classification	homo(term)	directed, unweighted, labeled	co-occurrence
21	classification	hetero(doc+concept)	undirected, weighted, unlabeled	bipartite graph
22	clustering	hetero(term+concept)	directed, unweighted, unlabeled	semantic tree
23	summarization	homo(sentence)	undirected, weighted, unlabeled	similarity
24	keyword extraction	hetero(term+concept)	directed, weighted, unlabeled	semantic tree
25	novelty detection	homo(term)	undirected, weighted, unlabeled	co-occurrence
26	opinion mining	hetero(term pair+doc)	undirected, unweighted, unlabeled	bipartite graph
27	search	homo(term)	undirected, weighted, unlabeled	co-occurrence
28	search	homo(term)	directed, weighted, labeled	co-occurrence

homo(homogeneous), hetero(heterogeneous), doc(document), Pos(Part of Speech),

Acknowledgements

This research was financially supported by Hansung University during sabbatical year(2013) for Jae-Young Chang. Also, this work was financially supported by Hansung University for Il-Min Kim.

References

[1] G. Salton, A. Wong and C. S. Yang, “A Vector Space Model for Automatic Indexing, Communications of the ACM”, vol. 18, no. 11, **(1975)**, pp. 613–620.

[2] http://en.wikipedia.org/wiki/Vector_space_model.

[3] J. Wu, Z. Xuan and D. Pan, “Enhancing Text Representation for Classification Tasks with Semantic Graph Structures”, International Journal if Innovative Computing, Information Control, vol. 7, no. 5(B), **(2011)**, pp. 2689-2698.

[4] K. M. Hammouda and M. S. Kamel, “Document Similarity Using a Phrase Indexing Graph Model”, Knowledge and Information Systems, vol. 6, no. 6, **(2006)**, pp. 710-727.

[5] S. Hensman, “Construction of Conceptual Graph Representation of Texts”, Proceedings of the Student Research Workshop at HLT-NAACL, **(2004)**, pp. 49-54.

[6] W. Wang, D. B. Do and X. Lin, “Term Graph Model for Text Classification”, Proceedings of the First international conference on Advanced Data Mining and Applications, **(2005)**, pp. 19-30.

[7] K. Valle and P. Ozturk, “Graph-Based Representation for Text Classification”, India-Norway Workshop on Web Concepts and Technologies, **(2011)**.

[8] M. Litvak and M. Last, “Graph-Based Keyword Extraction for Single-Document Summarization”, Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, **(2008)**, pp. 17-24.

[9] C. Jiang F. Coenen, R. Sanderson and M. Zito, “Text Classification Using Graph Mining-Based Feature Extraction”, Knowledge-Based Systems, vol. 23, no. 4, **(2009)**, pp. 302-308.

[10] J. Leskovec, M. Grobelnik and N. Milic-Fraying, “Learning Semantic Graph Mapping for Document Summarization”, Proceedings of the ECML/PKDD-2004 Workshop on Knowledge Discovery and Ontologies, **(2005)**.

[11] L. Zhang, C. Li, J. Liu and H. Wang, “Graph-Based Text Similarity Measurement by Exploiting Wikipedia as Background Knowledge”, World Academy of Science, Engineering and Technology, vol. 59, **(2011)**, pp. 1548-1553.

[12] G. Erkan and D. R. Radev, “LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization”, Journal of Artificial Intelligence Research, vol. 22, no. 1, **(2004)**, pp. 457-479.

[13] Y. Wu, Q. Zhang X. Huang and L Wu, “Structural Opinion Mining for Graph-based Sentiment Representation”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, **(2011)**, pp. 1332-1341.

[14] X. Wan and J. Yang, “Improved Affinity Graph Based Multi-Document Summarization”, Proceedings of the Human Language Technology Conference of the NAACL, **(2006)**, pp. 181-184.

[15] I. Yoo, X. Hu and I.-Y. Song, “Integration of Semantic-based Bipartite Graph Representation and Mutual Refinement Strategy for Biomedical Literature Clustering”, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, **(2006)**, pp. 791-796.

[16] R. Mihalcea, “Graph-Based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization”, Proceedings of 3rd International Conference on Emerging Trends in Engineering and Technology(ICETET), **(2010)**, pp. 516-519.

[17] R. Mihalcea and P. Tarau, “A Language Independent Algorithm for Single and Multiple Document Summarization”, Proceedings of International Joint Conference on Natural Language Processing, **(2005)**.

[18] R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Texts”, Proceedings of International Conference on Empirical Methods in Natural Language Processing, **(2004)**.

[19] S. Brin and L. Page, “The Anatomy of a Large-scale Hyper-textual Web Search Engine”, Proceedings of the seventh International Conference on World Wide Web 7, **(1998)**, pp. 107-117.

[20] A. Schenker, M. Last, H. Bunke and A. Kandel, “Classification of Web Documents Using a Graph Model”, Proceedings of Seventh International Conference on Document Analysis and Recognition, **(2003)**, pp. 240-244.

[21] R. Chau, A. C. Tsoi, M. Hagenbuchner and V. C. S. Lee, “A Concept Graph for Text Structure Mining”, Proceedings of the Thirty-Second Australasian Conference on Computer Science, vol. 91, **(2009)**, pp. 141-150.

Copyright ⓒ 2014 SERSC

[22] M. S. Hossain and R. A. Angryk, “GDClust: A Graph-Based Document Clustering Technique”, Proceedings of Seventh IEEE International Conference on Data Mining Workshops, **(2007)**, pp. 417-422.

[23] S. Hariharan and R. Srinivasan, “Studies on Graph based Approaches for Single and Multi-Document Summarizations”, International Journal of Computer Theory and Engineering, vol. 1, no. 5, **(2009)**, pp. 1793-8201.

[24] C. A. Chahine, N. Chaignaud, J. H. P. Kotowicz and J. P. Pecuchet, “Context and Keyword Extraction in Plain Text Using a Graph Representation”, Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, **(2008)**, pp. 692-696.

[25] M. Gamon, “Graph-Based Text Representation for Novelty Detection”, Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing, **(2006)**, pp. 17-24.

[26] B. Li, L. Zhou, S. Feng and K.-F. Wong, “A Unified Graph Model for Sentence-Based Opinion Retrieval”, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, **(2010)**, pp. 1367-1375.

[27] J. Tomita, H. Nakawatase and M. Ishii, “Graph-Based Text Database for Knowledge Discovery”, Proceedings of the 13th international World Wide Web conference, **(2004)**, pp. 454-455.

[28] F. Zhou, F. Zhang and B. Yang, “Graph-Based Text Representation Model and its Realization”, Proceedings of International Conference on Natural Language Processing and Knowledge Engineering, **(2010)**, pp. 1-8.

[29] J. M. Kleinberg, “Authoritative Sources in a Hyperlinked Environment”, Journal of ACM, vol. 45, no. 5, **(1999)**, pp. 605-632.

Authors

Jae-young Chang, received the B.S., M.S. and Ph. D degrees in Computer Science and Statistics from Seoul National University, Seoul, Korea in 1992, 1994 and 1999 respectively. He is currently a professor at Department of Computer Engineering, Hansung University, South Korea. His current research interests include database and data mining.

Ilmin Kim, received a B.S. degree in Computer Engineering from KyungBuk National University in 1984 and a Ph. D degree from Arizona State University. He is currently a Professor at Department of Computer Engineering, Hansung University, South Korea. His current Research interests include distributed computing and operating systems.

Copyright ⓒ 2014 SERSC

Copyright ⓒ 2014 SERSC