Document Mining Based on Semantic Understanding of Text

Khaled Shaban, Otman Basir, and Mohamed Kamel

Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
{kshaban, obasir, mkamel}@uwaterloo.ca

Abstract. This paper presents a new paradigm for mining documents by exploiting the semantic information of their texts. A formal semantic representation of linguistic inputs is introduced and utilized to build a semantic representation for documents. The representation is constructed through accumulation of syntactic and semantic analysis outputs. A new distance measure is developed to determine the similarities between contents of documents. The measure is based on inexact matching of attributed trees. It involves the computation of all distinct similarity common sub-trees, and can be computed efficiently. It is believed that the proposed representation along with the proposed similarity measure will enable more effective document mining processes. The proposed techniques to mine documents were implemented as components in a mining system. A case study of semantic document clustering is presented to demonstrate the working and the efficacy of the framework. Experimental work is reported, and its results are presented and analyzed.

Keywords: Document mining, semantic understanding, text representation, similarity measure, document clustering.

1 Introduction

As the sheer number of documents that are available online grows exponentially, the need to manage these documents also grows. The result of this exponential growth is what has become known as the *information overload* problem. Taking into consideration only the volume of information available via the Internet and the World Wide Web (WWW) presents a non-trivial real problem. The extent of this problem is immediately apparent to anyone who has tapped into the WWW, and attempted to locate specific desired information. Solving this problem involves processes such as *information gathering*, *information filtering* (IF), *information retrieval* (IR), *information extraction* (IE), *document classification*, and *document clustering*. The goal of these processes is to help users to have better access to documents that satisfy their information needs. The needs can be to discover or derive new information from documents, to find patterns across documents, and to separate the desired information from the noise. These computational processes constitute cornerstone tasks in the ever-developing study field of *document mining*.

The goal of this work is to introduce, build, and demonstrate a novel document mining system that is based on semantic understanding of the document contents. The

system is composed of components that facilitate semantic analysis of text, measuring similarity between documents, and applying the different mining processes. The aim is to show the working of the method and how this system can provide better performance than it is possible otherwise.

There are three aspects pertain to document mining approaches: a) Representation models. b) Similarity measures. c) Mining processes.

Representation Models: Conventional text representation models focus on whether a document contains specific keywords, and their appearance frequencies. For example, in the vector space model (VSM) [5, 21, 22], documents are represented by vectors containing the frequency of all possible words (*features*) in a document set. Since many words rarely occur in a particular document, many of these features will have zero or low frequencies. Therefore, features are selected according to their importance as dictated by criteria such as *Document Frequency-Inverse Document Frequency*, *Information Gain*, *Mutual Information*, a *Chi-Square-test*, and *Term Strength* [1, 29]. Moreover, before applying feature selection, a common practice is to reprocess text by removing stop-words and applying word-stemming algorithms. Stop-words, such as *the*, *and*, and *a*, are believed to have no significance in capturing meaningful information. Word-stemming algorithms convert different word forms into a similar canonical form. Two popular stemming algorithms are used; the Porter stemmer [20], and using a lexicon dictionary lookup, such as WordNet [18].

Despite the widespread use of these word-based approaches to represent documents, it is believed that these approaches contribute to the lack of reliable performance of document mining systems. These approaches consider the document as a bag of words, and ignore meanings and ideas the document wanted to convey. It is this deficiency that causes the similarity measures to either fail to perceive contextual similarity of text passages due to the variation of words the passages contain, or perceive contextually dissimilar text passages as being similar because of the resemblance of words the passages happen to have. An illustration of this deficiency that causes the failure is the two sentences '*John eats the apple standing beside the tree*' and '*The apple tree stands beside John's house*', where despite using the same words, their meanings are different. Moreover, the following two sentences have more or less the same meaning but have been constructed from different sets of words, '*John is an intelligent boy*', and '*John is a brilliant lad*'.

Similarity Measures: Similarity measures are used to determine distances between documents, after transforming the textual data into a useable and intelligible format. There are various techniques to measure similarities and they all rely on the chosen model to represent the text. In VSM, for instance, the feature space constitutes a geometric space where documents are represented as points in a multidimensional space. Thus, measuring the similarity can be easily calculated. Two measures are often used; they are the *cosine measure*, and the *Jaccard measure* [12, 25]. The cosine measure is defined as: $\cos(x,y)=(x.y)/(|x||y|)$, where $(x.y)$ denotes the vector dot product of x and y , and $|x|$ and $|y|$ are the lengths of vectors x , y , respectively. The cosine measure gives high similarity values to documents that share the same set of words with high term frequencies, and lower values to those that do not. The Jaccard measure is defined as: $\text{sim}(x,y)=|x \cap y|/|x \cup y|$. It finds the overlap between two

documents by calculating the number of terms that are common between them. The Jaccard measure can work for both continuous and binary feature vectors.

Although these techniques could be appropriate to measure distances between vectors of numerical features, the inherent inadequacy of word-focused approaches still exist. There is no perfect correlation between comparing words and comparing meanings. As the example sentences in the previous section show, the correlation can be very low indeed. Better mining results can be attained by measuring meanings instead of counting words found in documents. Thus it is imperative to develop semantic representations of text, and distance measures that can determine whether one statement is an instance of, or quite different from, another statement.

Mining Processes: Document mining processes may vary in their requirements and specifications, yet their goals are to discover and extract knowledge from documents. IR is concerned with finding relevant documents in response to a user request and ranking them accordingly [2, 17]. This is normally done by measuring the distances between documents and queries in their transformed form in an index. When relevancy and similarity measuring is performed with the intent of transmitting a document to a user(s), it is usually referred to as IF [3, 19]. It is also used to either accept or reject an incoming document, as in e-mail filters that attempt to screen for junk mail. The goal of IE is to locate specific information and produce structured format from unstructured or semi-structured documents [24]. The output of an extraction system is usually tabular or fixed-format forms that are filled out with unambiguous data. This is done through analyzing those portions of each document that contain relevant information. Relevance here is determined by predefined domain guidelines which specify what types of information the system is expected to find. The aim of the document classification task is to assign a new document to one of a pre-existing set of document classes. In this setting, the task of creating a classifier consists of discovering useful characterization of the documents that belong to each class. Although this can be done by hand, the standard approach is to use supervised machine learning. In particular, classifiers can be trained on a set of documents that have been labeled with the correct class. The classification task assumes existing categories, or clusters, of documents. By contrast, the task of document clustering is to create, or discover, a reasonable set of clusters for a given set of documents. A reasonable cluster is defined as one that maximizes the within-cluster document similarity, and minimizes between cluster similarities [4].

2 The Semantic-Based Document Mining System

The proposed approach is based on analyzing text in documents before proceeding with the different mining processes requirements. The text analysis step comprises syntactic analysis to extract syntax structural descriptions (e.g. part of speech tags, phrasal chunks, and parse trees), and semantic analysis that produces formal knowledge representations of the documents contents. Thus, a generic semantic-based document mining system consists of three major components: (1) Text parser, (2) Similarity estimator, and (3) Mining processes. These components are interconnected, which makes the system highly integrated, but yet modular.

2.1 Semantic-Based Text Representation

We introduce a representation, called the *Semantic Graph Model* (SGM), where the focus is on the ability to express distinct readings of sentences as distinct formulas that capture their intuitive structures and meanings. We start by creating predicate structures of sentences, augmenting the structures elements with valuable attributes, and taking all parsed sentences as the document representation (Figure 1). The representation is a graph-based data structure where entities, such as *agents*, *objects*, *states*, *actions*, *events*, and *locations* are represented as vertices, and relations between them are represented as arcs. Each node holds information that could include its original text, syntactic information, semantic meaning, and relations with other nodes.

Fig. 1. SGM Skeleton for a Document

2.2 Semantic-Based Similarity Measure

The similarity estimator component is responsible for searching the abstract representations of two documents, finding elements that are sufficiently similar, and yielding an overall similarity index. An inexact graph matching technique to approximately match graphs is utilized to calculate semantic distances between documents. Since SGMs are in essence trees with multiple symbolic node attributes, the semantic distance measure is defined over attributed-trees. The measure involves the computation of all similarity common sub-trees that do not overlap.

Figures 2 (a) and (b) demonstrate the proposed distance estimator. The figures depict the finding of the common similarity sub-trees in the SGMs of two sentences. The symbol (x) denotes the nodes that are not matched, while ($\Leftrightarrow$) represents the similarity matching of nodes that are included in the common similarity sub-trees. Note that in order to match only distinct sub-trees the overlapping resulting from considering one node in Figure 2 (a) (status: 'stands beside John's house') should be eliminated. Thus, only one matching node (Action 2 or Object 2) would be considered as a match.

Clearly, the summing up of similarities for the first pair of sentences is lower than the second pair, which reflects the closeness of their meaning distances. An index number indicating the similarity between a pair of sentences (or documents) can be determined by normalizing the similarity of all found similarity sub-trees. Applying this on the sample sentences in figure 2 (a) and (b), assuming the similarity values between the sub-trees are estimated through matching of nodes attributes will produce a low similarity for the first sentence pair an a high similarity for the second pair.

Fig. 2. (a) Similarity Estimation for Sample Texts. (b) Similarity Estimation for Sample Texts.

2.3 Semantic-Based Mining Processes

At a certain level of abstraction we can look at the different mining processes as being relatively similar. We maintain that they can all benefit from the semantic-based representation, i.e., the SGM, produced by a text parser, and can make use of the similarity measure estimator as needed. And then proceed with their specific tasks. The text analysis and the similarity estimation should be as much independent as possible of the specific mining process to allow for modularity and scalability. In Section 3, a case study of the proposed approach on one of these document mining processes (semantic-based document clustering) is presented. Implementations and experimental work have been carried out and results are also presented and analyzed.

3 Semantic Document Clustering: A Case Study

While a large number of statistical document clustering approaches have become available, relatively, a little attention has been paid to the clustering of documents represented using symbolic structures. In principle, however, given suitable documents similarity (or dissimilarity) measurements, many of the clustering algorithms originally developed in the context of statistical pattern recognition, can be applied in the symbolic domain. This is the course we have adapted to cluster documents represented semantically. Text documents were syntactically and semantically parsed and represented by their SGMs. Pair-wise similarities were computed using our developed inexact tree similarity measure. The similarity matrix was fed to different clustering algorithms to produce clusters for the data sets. The clustering results were then evaluated using standard evaluation techniques in document mining. In what follows we detail the steps taken and give an analytical discourse on findings.

Text Parser: A parsing system, which performs the syntactic analysis, and builds the semantic structures of SGM is implemented. We are utilizing the open source GATE (General Architecture for Text Engineering) project from University of Sheffield in the UK (<http://gate.ac.uk>) [11], and a commercial integrated development environment (IDE), Visual Text [26], from Text Analysis International, Inc. GATE provides an extensible framework for information extraction and text analysis, and Visual Text integrates NLP++ programming language for rapid parsers building. We have based the design of our text analysis system on a multi-pass, and a multi-strategy architecture that could be implemented within Visual Text IDE. The syntactic and semantic parser developments are built upon the TAIParse general analyzer [27] that is provided as an open source from Text Analysis. The analyzer contains 123 passes that build syntactic (parse trees), and semantic (the SGM) structures. Many other external processing components and language resources are wrapped and made usable interchangeably within the system. Specifically, every document in the processed data set goes through tokenization, part of speech tagging, syntactic parsing, semantics and discourse analysis.

Data Set: A Web documents collection is used. The document set is a collection of 2340 Reuters news articles posted on Yahoo! News. This corpus is especially interesting for evaluation, as it comes along with a *hand-crafted* classification. All documents have been classified manually by Yahoo experts to one or more of six main categories of Reuter's news feed, namely, business, entertainment, health, politics, sports, and technology. The data set has been collected and used by Boley et al [6, 7, 8] for clustering.

Text Representations: All text documents are represented using the proposed semantic graph model (SGM). The SGM representation model constitutes trees that have attributes for their nodes. Each document is represented as a tree rooted in a node describing its file name, and branches to sub-trees represent paragraphs, sentences, clauses, and predicate-argument nodes with attributes describing the arguments. As applicable, the node attributes include *Name*: a unique identification for the node, *Type*: the semantic role of concept, *Text*: the original text, *Syntax*: the

part of speech tag, *Synonyms*: dictionary senses of the concept, *Semantic*: the disambiguated meaning of the concept, and *Relation*: other nodes IDs that are connected to the node. Different algorithms are employed and integrated in the parser to produce the SGM representations. The documents are also represented by the VSM. Some pre-processing procedures are performed such as normalizing words, analyzing words globally, and words weighting. To normalize words, numeric, and stop words are removed. All words are converted to lowercase, and words of length 2 or less are also removed. Note that no stemming is done. Analyzing words globally is done by building a list of unique words from all documents, and calculating the document frequency of each word. Then, for each document, the word frequency is calculated and weighted using the following formula: $w_{i,j} = f_{i,j} \log(\frac{N}{df_j})$, where $f_{i,j}$ is

the term frequency of word j in document i , N is the number of documents, df_j is the document frequency of word j . Finally, an output of the term-by-document matrix is produced.

Similarity Measures: Our distance measure technique is used to measure distances between documents. It follows the similarity calculation approach that is based on finding all distinct common similarity sub-trees. The algorithm is applied to the semantic trees of the SGMs produced by the semantic parser. Various attributes are considered in the testing. Attribute consideration starts from taking into account only the original text to represent a node and extends to consider other attributes the parser accumulates. Specifically, we carried out the experiments with mainly five options of sets of attributes to include in measuring the similarities. These options are (1) original text only, (2) original text, and syntax tag, (3) original text, syntax tag, and semantic role, (4) original text, syntax tag, semantic role, and semantic disambiguated sense, and (5) original text, syntax tag, semantic role, semantic disambiguated sense, and semantic variation. To measure similarities of documents represented in the vector space, the commonly used cosine correlation measure is utilized.

Clustering Algorithms: The semantic approach to represent text and determine similarity between them is evaluated by allowing clustering algorithms to use the produced similarity matrix of the document set. We run the test with standard clustering algorithms that accept the pair-wise (dis)similarity between documents, rather than the internal vector-space or SGM representations of the documents. The employed algorithms included: Single-Pass Clustering [10, 15], k-nearest neighbour (k -NN) Clustering [10, 13], and Hierarchical Agglomerative Clustering (HAC) [16]. The different parameters of these algorithms (i.e., the clustering threshold parameters) are tuned so the best results of clustering are obtained.

Evaluation: To evaluate results, we used a benchmark of manually classified document set; Routers news feeds. The main aspect of the evaluation is the quality of the clustering task output, which is measured in terms of clusters quality. The widely used evaluation indices, F-measure that combines precision (P) and recall (R) [9, 14, 28], Entropy, and Overall-Similarity are used for this purpose as clustering quality evaluation measures. R for cluster c is the ratio of processed documents manually classified as c with regard to the whole document set, and P is the ratio of these documents clustered as c by the process that were also manually classified as c

regarding the produced set of results. The *F-measure* is calculated as: $F=[(\beta^2+1)PR]/[\beta^2P+R]$, the parameter β influences how much to favor recall over precision. The *F1* score where $\beta=1$, weighs precision and recall equally. The entropy E_i of a cluster i is calculated using Shanon [23] standard formula: $E_i = -\sum_j P_{ij} \log(P_{ij})$,

where P_{ij} is the probability of documents of cluster i belong to class j . An overall entropy E for all clusters can also be calculated as the sum of entropies for each cluster weighted by the size of each cluster as follows: $E = \sum_i (\frac{N_i}{N} \times E_i)$, where N_i is the

size of cluster i , and N is the total number of documents. Overall similarity uses the weighted similarity of the internal cluster similarity as: $\text{OverallSimilarity}(S) = \frac{1}{|S|^2} \sum_{x,y \in S} \text{sim}(x,y)$, where S is the cluster under consideration, and $\text{sim}(x,y)$ is the similarity between the two objects x and y .

3.1 Analysis of Results

The clustering algorithms were utilized to cluster the data set using document-to-document similarities produced by the semantic-based approach, and the vector space model. Results show the effectiveness of the semantic-based approach for clustering. In addition, the experiments illustrate improvements when more semantic clues are included in the process of measuring similarity between SGMs of documents. The improvements were achieved at a factor up to 72% from the base case of considering only the original text of nodes (option 1) to considering all five semantic analysis results (option 5). The threshold parameters for the different clustering algorithms were manually tuned and the ones that produced the best results were reported. The *F-measure* index showed noticeable improvement. The same can be noticed about the entropy and the overall similarity indexes.

Fig. 3. Clustering Results

To better understand the effect of the inclusion of semantic information when calculating similarity on the clustering quality, we plot the clustering quality profile indices against the similarity options in Figure 3. The plotted values are the averages of *F*-measure, entropy, and overall similarity of the different clustering algorithms. We also plot the value of these measures with regard to the VSM. It is easy to notice the enhancement of the clustering as we consider more semantic information. The enhancement is, however, non-linear. Options 2 and 3 seem not to have as much effect on the clustering quality, but they never bring it down. Option 5 had the most effect to bring up the quality of clustering. As anticipated; keywords alone cannot capture all the similarity information between documents, thus the quality of semantic clustering with option 1 was as low as the VSM.

4 Conclusions

We demonstrated a system composed of semantic components in an attempt to improve the accuracy of measuring the similarity between documents and using the similarity in applying document mining. We considered semantic document clustering as a case study. By exploiting the semantic analysis findings we could achieve better clustering results. We implemented document analysis components that are capable of identifying the meaning structures of text in documents. The second part is to measure similarity between parsed documents. The measure has the most impact on the performance as how much of semantic information it considers. Using the maximum amount of the semantic information enables us to perform similarity calculations between documents in a very robust and accurate way. The quality of clustering achieved using this model significantly surpasses the traditional vector space model based approach.

References

1. Aas, K., and Eikvil, L., "Text categorisation: A survey," Tech. Report 941, Norwegian Computing Center, 1999.
2. Baeza-Yates, R., and Ribeiro-Neto, B., *Modern Information Retrieval*, ACM, 1999.
3. Belkin, N. J., and Croft, W. B., "Information filtering and information retrieval: Two sides of the same coin?" *Comm. of the ACM*, vol. 35, no. 12, pp. 29-38, 1992.
4. Berkhin, P., "Survey of Clustering Data Mining Techniques," Tech. Report, Accrue Software, 2002.
5. Berry, M. W., Dunais, S. T., and O'Brien, G. W., "Using Linear Algebra for Intelligent Information Retrieval," *SIAM Review* 37(4), pp. 573-595, 1995.
6. Boley, D., "Principal direction divisive partitioning", *Data Mining and Knowledge Discovery*, 2(4):325-344, 1998.
7. Boley, D., Gini, M., Gross, R., Han, S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J., "Partitioning-based clustering for web document categorization", *Decision Support Systems*, 27:329-341, 1999.
8. Boley, D., Gini, M., Gross, R., Han, S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J., "Document categorization and query generation on the World Wide Web using WebACE", *AI Review*, 13(5-6):365-391, 1999.

9. Chakrabarti, S., *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003.
10. Cios, K., Pedrycs, W., and Swiniarski, R., *Data Mining Methods for Knowledge Discovery*. Kluwer Academic Publishers, 1998.
11. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications" *Proce. of 40th Anniversary Meeting of the ACL (ACL'02)*. Philadelphia, July 2002.
12. Cutting, D.; Karger, D.; Pedersen, J.; and Tukey, J.: "Scatter/gather: A cluster-based approach to browsing large document collections." In *16th International ACM SIGIR Conference on Research and Development in IR*, pp 126-135, 1993.
13. Dasarathy, B.V. "Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques." McGraw-Hill Comp. Scie. Series. IEEE Comp. Soci. Press, 1991.
14. Eikvil, L., "Information Extraction from World Wide Web – A Survey," Technical Report 945, Norwegian Computing Center, July, 1999.
15. Hill, D.R. "A vector clustering technique" In Samuelson, ed. *Mechanized Information Storage, Retrieval and Dissemination*. North-Holland, (1968).
16. Jain, A. K., and Dubes, R. C., "Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs NJ, U.S.A, 1988.
17. Lee, D. L., Chuang, H., and Seamons, K., "Document Ranking and the Vector-Space Model" *IEEE Comp., Issues on Assessing Measurement*, 14(2):67-75, 1997.
18. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K., "Introduction to WordNet: An On-line Lexical Database" *Cognitive Science Lab*, Princeton University, 1993.
19. Mostafa, J., Mukhopadhyay, S., Lam, W., Palakal, M., "A Multi-level Approach to Intelligent Information Filtering: Model, System & Evaluation," *ACM Transactions on Information Systems*, 15(4): pp. 368-399, 1997.
20. Porter, M. F. "An algorithm for suffix stripping" *Program*, 14(3), pp.130-137, 1980.
21. Salton, G., and McGill, M. J., *Introduction to Modern Information Retrieval*, McGraw-Hill, 1984.
22. Salton, G., Wong, A., and Yang, C., "A vector space model for automatic indexing," *Communications of the ACM*, 18(11), pp.613-620, 1975.
23. Shannon, C.E.: "A Mathematical Theory of Communication", *Bell Syst. Tech. J.*, 27, 379-423, 623-656, 1948.
24. Soderland, S., *Learning Information Extraction Rules for Semistructured and Free Text*. Machine Learning, 1999.
25. Steinbach, M.; Karypis, G.; and Kumar, V. "A comparison of document clustering techniques" *KDD-2000 Workshop on TextMining*, August 2000.
26. Text Analysis International, Inc. "Integrated Development Environments for Natural Language Processing", White Paper, www.textanalysis.com, October 2001.
27. Text Analysis International, Inc. "Multi-Pass Multi-Strategy NLP", White Paper, <http://www.textanalysis.com/>, October 2003.
28. van Rijsbergen, C. J.: *Information Retrieval*, Woburn Massachusetts. Butterworths, 1979.
29. Yang, Y., and Pedersen, J., "A Comparative Study on Feature Selection in Text Categorization" In *Proceeding of the 14th International Conference on Machine Learning*, ICML, pp. 412-420, Nashville, TN, 1997.