Representing Document as Dependency Graph for Document Clustering

Yujing Wang[1]*, Xiaochuan Ni[2], Jian-Tao Sun[2], Yunhai Tong[1], Zheng Chen[2]

[1]Key Laboratory of Machine Perception
Peking University
Beijing 100871, P. R. China
{wangyj, tongyh}@cis.pku.edu.cn

[2]Microsoft Research Asia
No.5 Danling Street, Haidian District
Beijing 100080, P. R. China
{xini, jtsun, zhengc}@microsoft.com

ABSTRACT

In traditional clustering methods, a document is often represented as "bag of words" (in BOW model) or n-grams (in suffix tree document model) without considering the natural language relationships between the words. In this paper, we propose a novel approach DGDC (Dependency Graph-based Document Clustering algorithm) to address this issue. In our algorithm, each document is represented as a dependency graph where the nodes correspond to words which can be seen as meta-descriptions of the document; whereas the edges stand for the relations between pairs of words. A new similarity measure is proposed to compute the pairwise similarity of documents based on their corresponding dependency graphs. By applying the new similarity measure in the *Group-average* Agglomerative Hierarchial Clustering (GAHC) algorithm, the final clusters of documents can be obtained. The experiments were carried out on five public document datasets. The empirical results have indicated that the DGDC algorithm can achieve better performance in document clustering tasks compared with other approaches based on the BOW model and suffix tree document model.

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering - *algorithms, similarity measures*; H.3.1 [**Content Analysis and Indexing**]: Linguistic processing; I.2.7 [**Artificial Intelligence**]: Natural Language Processing - *text analysis*

General Terms

Algorithms, Experimentation

Keywords

Document Clustering, Dependency Graph, Document Representation Model, Similarity Measure

1. INTRODUCTION

Document clustering techniques usually rely on four modules: document representation model, similarity measure, clustering model and the clustering algorithm which generates clusters based on the document representation model [5]. Among all these modules, the document representation model is very fundamental and crucial for the clustering results.

The most basic model for document representation is the Vector Space Document (VSD) model. In this model, the document is regarded as "bag of words" (BOW), without considering the relationships between the words. In order to achieve better clustering results, many efforts have been made to seek more informative document representation models. The n-gram model [12] is one of those efforts, which can be viewed as an extension of BOW model but includes all ordered sequences of less than n words in the feature vector. The suffix tree document model [14] considers a document to be a set of suffix substrings and constructs a suffix tree using substrings of all documents in the corpus. It provides a flexible n-gram approach by identifying all overlapping phrases among documents as Longest Common Prefixes (LCPs) [9]. Some research works [7, 6, 13] also leveraged ontologies to enrich the representation of documents. WordNet [6], Mesh [13] and Wikipedia [7] have been adopted and improvements were achieved in document clustering tasks. However, the pairwise relationships of words which are suggested in the natural language sentences are still ignored in these document representation models.

In this paper, we propose a more informative document representation model, namely the Dependency Graph-based Document (DGD) model. In this model, each document is represented as a dependency graph where each node corresponds to a word which can be seen as a meta-description of the document. The edges between nodes are used to catch the semantic relations between word pairs. A novel similarity measure is also proposed for calculating the similarity of documents based on their corresponding dependency graphs. The Dependency Graph-based Document Clustering (DGD-C) algorithm is conducted in the following steps: (1) Construct a dependency graph for each document. (2) Calculate

[*] This work is done when the first author is visiting Microsoft Research Asia.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.
Copyright 2011 ACM 978-1-4503-0717-8/11/10 ...$10.00.

the similarity of each pair of documents based on the novel similarity measure. (3) Generate the final clusters of documents by the *Group-average* Agglomerative Hierarchical Clustering (GAHC) algorithm [14]. The experiments were carried out on five public document datasets. The empirical results have indicated that the DGDC algorithm can achieve better performance in document clustering tasks compared with other approaches based on the BOW model and the suffix tree document model.

2. DEPENDENCY GRAPH-BASED DOCUMENT CLUSTERING ALGORITHM

2.1 Dependency Graph-based Document Model

Suppose d is a document with a vocabulary set $W = \{w_1, w_2, ..., w_n\}$, where w_i stands for a word which appears in d and $w_i \neq w_j$ for $\forall i \neq j$. Since stopwords (e.g., the, is) count little for the meaning of the whole document, we exclude them from the vocabulary set. The dependency graph G corresponding to document d is denoted as $G = (V, E)$. $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices in the graph, where each vertex v_i corresponds to w_i in the vocabulary set. $E = \{e_1, e_2, ..., e_m\}$ is the collection of edges, where each edge e_j is associated to a pair of vertices, which indicates that there is some relationship between them.

Consider a document A, which has the content "Beijing is a big city. The city is very beautiful". Figure 1 shows the dependency graph corresponding to the document. Assume that we have two documents A and B. A is "Beijing is a big city. The city is very beautiful". B is "Beijing is a very beautiful city which is big". These two documents have the same semantic meaning so that they should have high similarity between each other. However, as they are literally organized by different word sequences, the similarity is much lower than expected according to the n-gram model and the suffix tree document model. In our model, document A and B correspond to the same dependency graph which correctly indicates that they are semantically equal with each other.

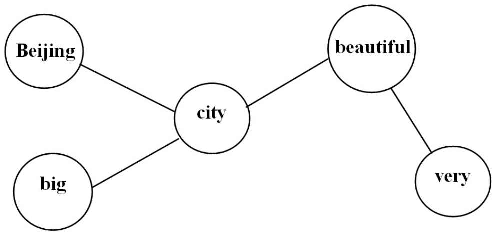

Figure 1: The dependency graph generated for "Beijing is a big city. The city is very beautiful."

In practice, a dependency parser (e.g. the Stanford parser [2]) is needed to obtain word relations from the original sentences. Using the parser, the dependency graph can be constructed in the following steps. Initially, there are no vertices and edges in the graph. Then the vertices and edges are added by processing each sentence in the document sequentially. For each sentence, we parse it using the dependency parser, which outputs a set of words and the identified pairwise relations between them. The non-stopwords are then added to the vertex set if they are not contained in the graph before. Thus, each non-stopword in the output is associated with one vertex in the graph. Finally, for each pairwise relation suggested by the parser, a new edge will be generated between the corresponding vertices if it does not exist in the graph.

2.2 Similarity Measure

Given a dependency graph $G = (V, E)$, suppose the vertex is denoted by v_i and w_i is the word corresponding to v_i. The weight of vertex v_i associated to document d can be calculated by the traditional *tf-idf* measure.

After representing the documents as weighted dependency graphs, the similarity of two documents can be calculated based on their corresponding graphs. Suppose there are two documents d_1 and d_2 in the global corpus D, whose corresponding graphs are $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ respectively. $V_1 \bigcup V_2$ is the collection of nodes contained in G_1 and G_2, where the nodes corresponding to the same word are considered the same. The feature weight matrix R_1 for document d_1 is defined as $R_1 = \{r_{ij} | 0 \leq i, j < |V_1 \bigcup V_2|\}$. When $i = j$, r_{ij} is actually the weight of single word feature denoted by node v_i. When $i \neq j$, r_{ij} measures the importance of the relation between v_i and v_j.

Mathematically, r_{ij} is defined as:

$$r_{ij} = \begin{cases} c(v_i, d_1) \cdot \beta & \text{if } i = j \\ \sqrt{c(v_i, d_1) \cdot c(v_j, d_1)} \cdot f_{G_1}(v_i, v_j) & \text{otherwise} \end{cases} \quad (1)$$

where $c(v_i, d_1)$ is the weight of vertex v_i associated to document d_1. β is a constant coefficient to balance single word features (when $i = j$) and relation features (when $i \neq j$). $f_{G_1}(v_i, v_j)$ is a penalty function which depicts how tightly the vertex v_i and v_j are connected with each other in G_1. It can be calculated by:

$$f_{G_1}(v_i, v_j) = \begin{cases} 0 & \text{if } v_i \notin V_1 \text{ or } v_j \notin V_1 \\ (\frac{1}{dist_{G_1}(v_i, v_j)})^{\alpha} & \text{otherwise} \end{cases} \quad (2)$$

where $dist_{G_1}(v_i, v_j)$ is the distance (length of the shortest path) between v_i and v_j in G_1, α is a penalty coefficient. The lengths of all the edges in the graph are set to 1.

Let $R_1 = \{r_1(i, j)\}$ and $R_2 = \{r_2(i, j)\}$, The similarity of these two feature weight matrixes is computed by

$$Sim(R_1, R_2) = \sum_{i \leq j} r_1(i, j) \cdot r_2(i, j) \quad (3)$$

Consequently, the similarity of two documents d_1 and d_2 is defined as:

$$Sim(d_1, d_2) = \frac{Sim(R_1, R_2)}{\sqrt{Sim(R_1, R_1) \cdot Sim(R_2, R_2)}} \quad (4)$$

such that $Sim(d_1, d_2) \in [0, 1]$.

2.3 Clustering Algorithm

The *Group-average* Agglomerative Hierarchical Clustering (GAHC) algorithm is adopted in the cluster generating procedure. The algorithm considers each document as a unique cluster initially and selects a pair of clusters to merge repeatedly in the merging procedure. In each turn, the pair of most similar clusters are selected to be merged. The similarity of two clusters is calculated by the *group-average* measure.

One important issue for hierarchical clustering is to determine which step to terminate in the merging procedure [11]. One way to solve this problem is using a pre-defined number of clusters or setting a constant threshold of similarity for termination. However, neither of the strategies are practical for various kinds of corpus in applications. Motivated by [11], we leverage the R^2 criterion to determine the terminating point in the merging procedure.

The R^2 value represents the proportion of variance accounted for by the clusters, which is estimated by:

$$R^2 = 1 - \frac{\sum_k \sum_{d_i, d_j \in C_k} Dist(d_i, d_j)}{\sum_{d_i, d_j \in D} Dist(d_i, d_j)} \tag{5}$$

where C_k stands for the k^{th} cluster, $Dist(d_i, d_j) \in [0, 1]$ estimates the distance between document d_i and d_j. We define:

$$Dist(d_i, d_j) = 1 - Sim(d_i, d_j) \tag{6}$$

We can plot R^2 with respect to the number of remaining clusters in each step. The place where the curve levels off is selected to be the terminating point.

3. EXPERIMENTS

3.1 Experimental Setup

We select five public document datasets: *CSTR*, *Reuters3*, *News-diff3*, *News-sim3* and *News-mod6*, which are commonly used to evaluate document clustering and categorization methods. *CSTR*[1] consists of abstracts for technical reports. *Reuters3* is generated from the *Reuters-21578*[2] corpus. We choose three popular topics (grain, trade, and crude) and randomly select 100 documents for each topic. *News-diff3*, *News-sim3* and *News-mod6* datasets [1] are derived from the *20-Newsgroups* corpus[3]. The *News-diff3* dataset consists of three different newsgroups (alt.atheism, rec.sport.baseball, and sci.space) and the *News-sim3* dataset contains three similar newsgroups (comp.graphics, comp.os.ms-windows.misc, and comp.windows.x). The *News-mod6* dataset consists of 600 documents from 6 newsgroups (rec.sport.baseball, sci.space, alt.atheism, talk.politics.guns, comp.windows.x, and soc.religion.christian). In the *News-mod6* dataset, some topics are similar while others are different from each other. Table 1 summarizes the characteristics of these five datasets.

Table 1: Summary of datasets

Dataset	Number of document	Number of class	Average document length (by word)
CSTR	635	4	149.4
Reuters3	300	3	463.0
news-diff3	300	3	363.9
news-sim3	300	3	397.9
news-mod6	600	6	649.8

Three metrics are adopted in our experiments to evaluate the performance of document clustering: *normalized mutual information (NMI)* [4], *F-measure* [3] and *purity* [15].

In our experiments, the Stanford parser[4] [8] is used to obtain word dependencies. The Porter stemming algorithm [10] is utilized to stem the words in original documents. Instead of using a standard stopword list, we determine the stopwords by calculating the document frequency df of each word, similar to that in [3]. A word w_i is considered to be stopword if $\frac{df(w_i)}{|D|} > 0.5$, where $df(w_i)$ is the number of documents containing w_i in the global corpus and $|D|$ is the size of global corpus. The parameters in the similarity measure are set as $\alpha = 0.5$ and $\beta = 0.5$ experimentally.

The baseline approach utilizes the "bag of words" (BOW) model for document representation and weights each word in the feature vector by *tf-idf* measure. We also compare our algorithm with NSTC [3] method, which represents the documents as a suffix tree and performs clustering based on a suffix tree similarity measure. According to [3], the maximum length of Longest Common Prefixes (LCPs) in the suffix tree is set to be 5. To obtain a fair comparison between the DGDC algorithm and other approaches, we adopt the same way of word stemming and stopword recognition. Moreover, the *Group-average* Agglomerative Hierarchical Clustering algorithm is utilized to generate clusters in all the three approaches and the R^2 criterion is adopted to determine the cluster number.

3.2 Results

Firstly, we demonstrate the feasibility of using R^2 criterion to determine the actual stop point in the *Group-average* Agglomerative Hierarchical Clustering (GAHC) algorithm. An example is illustrated by applying the DGDC algorithm on *news-diff3* dataset. Figure 2 plots the R^2 value with respect to the number of remaining clusters in each step. It can be noticed that the curve levels off at point A (the number of clusters is 9 and the value of R^2 is 0.851) in the figure, so that the merging procedure will be terminated when the number of remaining clusters equals to 9. As shown in Figure 3, it is exactly the point where the best performance can be achieved. Therefore, the R^2 criterion provides an appropriate and practical way to determine the point of termination in the merging procedure.

Secondly, we compare our algorithm with other approaches. The clustering results of different algorithms on five datasets are shown in Table 2, measured by *NMI*, *F-measure* and *purity* respectively. The number of remaining clusters is also listed to indicate the place of termination. As shown in Table 2, the DGDC algorithm outperforms other approaches on all the five datasets.

4. CONCLUSION AND DISCUSSION

In this paper, we propose a novel document representation model. A new similarity measure for documents is also proposed, which calculates the similarity based on dependency graphs. The similarity measure is applied to the *Group-average* Agglomerative Hierarchical Clustering (GAHC) algorithm and promising results are obtained in the document clustering experiments.

[1] *http://www.cs.rochester.edu/trs*

[2] *http://www.daviddlewis.com/resources/testcollections/*

[3] http://people.csail.mit.edu/jrennie/20Newsgroups/

[4] The parser is available for download as open source at: http://nlp.stanford.edu/downloads/lex-parser.shtml

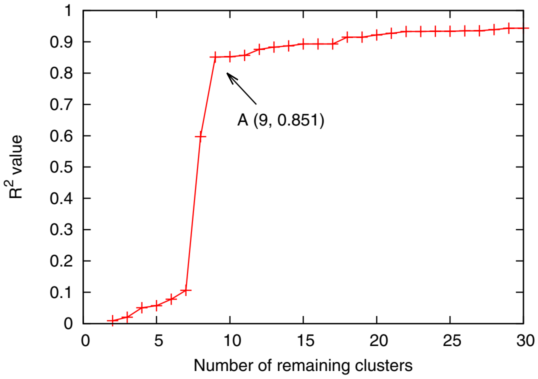

Figure 2: The plot of R^2 value with respect to the number of remaining clusters

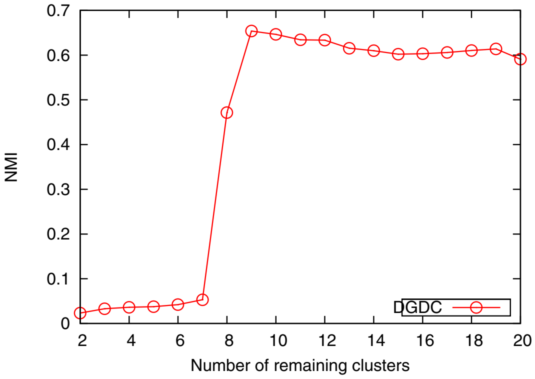

Figure 3: The plot of *NMI* value by DGDC algorithm on *news-diff3* dataset

Although good results have been achieved, we have to point out the limitation of the algorithm. Firstly, it takes $O(Nm^2 \log m + N^2m^2)$ to calculate all the pairwise similarities of documents, where N stands for the number of documents in the corpus and m represents the average word length of the documents. This high time complexity will make the algorithm computationally prohibitive for clustering tasks that deal with large corpus. Moreover, the R^2 criterion used in this approach for cluster number estimation is not fully automatic and sometimes needs manual efforts to help.

5. REFERENCES

[1] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *Proceedings of KDD'03*, pages 19–28.

[2] D. Cer, M.-C. de Marneffe, D. Jurafsky, and C. Manning. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of LREC'2010*, pages 1628–1632.

[3] H. Chim and X. Deng. A new suffix tree similarity measure for document clustering. In *Proceedings of WWW'07*, pages 218–225.

[4] B. E. Dom. An information-theoretic external cluster-validity measure. *Research Report RJ 10219, IBM*, 2001.

Table 2: Clustering results on five datasets

Algorithm	NMI	F-measure	Purity	Number of clusters
CSTR				
DGDC	**0.654**	**0.862**	**0.872**	10
NSTC	0.593	0.797	0.775	11
BOW	0.597	0.769	0.820	18
Reuters3				
DGDC	**0.592**	**0.778**	**0.897**	13
NSTC	0.468	0.671	0.850	10
BOW	0.532	0.706	0.887	13
News-diff3				
DGDC	**0.657**	**0.893**	0.903	10
NSTC	0.648	0.888	**0.913**	8
BOW	0.505	0.790	0.863	17
News-mod6				
DGDC	**0.767**	**0.834**	**0.810**	15
NSTC	0.738	0.819	0.795	16
BOW	0.734	0.814	0.807	20
News-sim3				
DGDC	**0.251**	**0.471**	**0.713**	16
NSTC	0.221	0.444	0.663	20
BOW	0.210	0.464	0.637	20

[5] K. M. Hammouda and M. S. Kamel. Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):1279–1296, 2004.

[6] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Proceedings of Semantic Web Workshop, SIGIR'03*.

[7] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of KDD'09*, pages 389–396.

[8] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003.

[9] U. manber and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[10] M. Porter. New models in probabilistic information retrieval. *British Library Research and development Report No.5587*, 1980.

[11] W. S. Sarle. Cubic clustering criterion. *SAS Technical Report A-108*, Cary, NC: SAS Institute Inc. 1980.

[12] A. Tomovic, P. Janicic, and V. KeŽelj. N-gram-based classification and unsupervised hierarchical clustering of genome sequences. *Computer Methods and Programs in Biomedicine*, 81(2):137–153, 2006.

[13] I. Yoo, X. Hu, and I.-Y. Song. Integration of semantic-based bipartite graph representation and mutual refinement strategy for biomedical literature clustering. In *Proceedings of KDD'06*, pages 791–796.

[14] O. zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of SIGIR'98*.

[15] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. *Technical Report, Department of Computer Science, University of Minnesota*, 2002.