A Significance-based Graph Model for Clustering Web Documents

Abstract. Traditional document clustering techniques rely on single-term analysis, such as the widely used Vector Space Model. However, recent approaches have emerged that are based on Graph Models and provide a more detailed description of document properties. In this work we present a novel Significance-based Graph Model for Web documents that introduces a sophisticated graph weighting method, based on significance evaluation of graph elements. We also define an associated similarity measure based on the maximum common subgraph between the graphs of the corresponding web documents. Experimental results on artificial and real document collections using well-known clustering algorithms indicate the effectiveness of the proposed approach.

1 Introduction

Web documents have a distinguished role in modern information society, therefore much research activity has focused on how to organize such information. The general objective of web document clustering methods is to automatically segregate documents into groups called clusters, in a way that each group represents a different topic and ideally includes all similar documents. The problem belongs to the Web Mining area and, more specifically, to Web Content Mining [1].

In order to perform clustering of Web documents two main issues must be addressed. The first is the definition of a representation model for Web documents along with a measure quantifying the similarity between two Web document models. Although single-term analysis is a simplified approach with limited capabilities, the Vector Space Model that relies on word counts (or frequencies) is still in wide use today. However, new approaches are emerging based on *graph representations* of documents which may be either *term-based* [2] or *path-based* [3]. A fundamental difference between graph approaches is the ability to utilize document representatives, instead of modeling all their information. The model we propose in this work uses document representatives of adjustable size and achieves great modeling performance, while conforming to computational effort conditions (CPU, memory, time).

The second issue in Web document clustering concerns the employment of a clustering algorithm that will take as input the similarity matrix for the pairs of documents and will provide the final partitioning. In this work we considered three clustering algorithms based on the hierarchical agglomerative approach and on the well-known k-means method.

This is the extended version of the work that was published as short paper (4 pages) in the Proceedings of SETN 2006, LNAI 3955, Springer-Verlag, pp. 516-519, 2006.

2 Web Document Analysis

Web documents are primarily HTML documents, where a set of tags is used to designate different document parts and thus assign layout or structural properties. The role of the analysis task is to locate the ‘useful’ information in a Web document, which is mostly written having in mind layout and style related issues. Thus, much structural information is omitted and is left to the visual understanding of the user. As a consequence, the following problems are encountered: i) HTML is a semi-structured language, ii) punctuation mark omissions, e.g., many sentences do not end up with a period.

Nevertheless, there is still sufficient information to successfully extract the document parts and evaluate their *significance levels*. The key observation that promises better modeling for web documents is that we can exploit the provided structural and layout properties to assign various importance levels to different document parts. The process is simple and is based on a predefined correspondence between HTML tags and significance levels. More specifically, we parse a document by iterating the steps below:

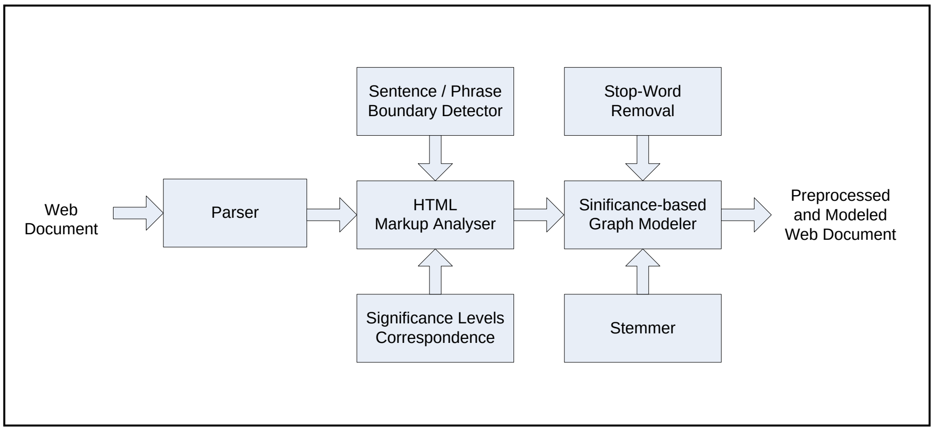
Sentence / Phrase Boundary Detector
Stop-Word Removal
Web Document
Parser
HTML Markup Analyser
Sinificance-based Graph Modeler
Preprocessed and Modeled Web Document
Significance Levels Correspondence
Stemmer

Fig.1. The preprocessing and modeling process of a document

1. Locate the beginning of next document part, by ignoring data irrelevant to our objective (scripts, HTML comments, irrelevant tags, etc).
2. Estimate the significance level of the current recognized document part.
3. For each existing term in the current document part we:
    3.1. apply a cleaning process, and
    3.2. update the general document significance value (see Sect. 3).
4. Locate the end boundary of current document part.

In our implementation four significance levels were used: {VERY HIGH, HIGH, MEDIUM, LOW}. Examples of document parts with very high significance are the title and metadata (description and keywords). High significance is assigned to section titles, medium to emphasized parts, and finally the lowest level is assigned to the remainder of normal text that has no special properties. As discussed in Section. 4, these levels are to be considered in the definition of the proposed similarity measure.

The tools that were used to alleviate difficulties in Web document analysis are a *meaning detector* that decides which parts of source files are useful for our task, and a *sentence/phrase boundary detector*, which locates parts of different significance. Both have been implemented using heuristic rules and result in an effective and correct system for the majority of Web documents we considered. Furthermore, a cleaning process was applied including a traditional stemming process [4] along with removal of stop words.

3 Significance-based Graph Representation

In this section, we present a graph model for Web documents, where document terms are represented as nodes and their semantic correlation as edges. This has been proved to be a successful representation [2] but non-weighted graphs were considered. Here, we propose a generalization of this model that provides *weighted graph* representations. The advantage is that it can describe more efficiently web document features and thus choose 'better' representative graphs for each document. Moreover, we can devise a new similarity measure that exploits the significance information.

We represent a document as a directed graph, well known as *DIG* (*Directed Indexed Graph*), along with a weighting scheme. Formally, a document $d = \{W, E, S\}$ consists of three sets of elements:

W: is a set of graph nodes $W = \{w_{1,}\ w_{2,}\ w_{3,\ \ldots,}\ w_{|W|}\}$ each of them uniquely represents a word of the document (unique node label in graph).

E: is a set of graph edges $E = \{e_{1,}\ e_{2,}\ e_{3,\ \ldots,}\ e_{|E|}\}$, where $e_i = (w_{k,}\ w_l)$ is an ordered pair of graph nodes denoting the existence of a directed edge from w_k to w_l. Indeed, we call w_l *neighbor* of w_k and the *neighborhood* of w_k is the set of all the neighbors of w_k. These properties capture semantic correlations between terms.

S: a function *S*, which assigns real numbers as significance weights to the *DIG* elements (nodes and edges).

3.1 Significance Weighting Model

The simplest weighting scheme is actually a non-weighting scheme (*NWM*) [2]. The next step is the assignment of frequencies as graph weights for nodes (*FM*), whereas in this work we propose a more sophisticated significance-based weighting scheme (*SM*). To define the scheme, we use the symbol g_w for node significance and g_e for the significance of edges. We define the node (term) significance g_w as:

$$g_w(w, d) = \sum_{i=1}^{freq(w,d)} s_i \,,$$

where d is a document, w is a document word, $freq(w, d)$ is the frequency of word w in document d, and s_i is the significance level of i-th occurrence of the word w (possible values are {VERY HIGH, HIGH, MEDIUM, LOW}).

Regarding to the edges, we should keep in mind the key role they have for document's meaning content, since they represent term associations. Thus, we define the edge significance g_e as a function of the significance of the respective terms as well as the edge frequency:

$$g_e(e(w_k, w_l), d) = \frac{g_w(w_k, d) \cdot g_w(w_l, d)}{g_w(w_k, d) + g_w(w_l, d)} \cdot freq(e(w_k, w_l), d),$$

where $e(w_k, w_l)$ is a document edge and $freq(e(w_k, w_l), d)$ is the edge's frequency in document d. We are now in a position to define the *document content*, which would be based on the weights of all elements of the document graph:

$$g_D^{(all)}(d) = \sum_{j=1}^{nodenum(d)} g_w(w_j, d) + \sum_{i=1}^{edgenum(d)} g_e(e_i(w_k, w_l), d),$$

where $nodenum(d)$ and $edgenum(d)$ are the number of different words and edges respectively in document d.

3.2 Construction of Representative Graphs – Term Filtering

As mentioned in the introduction, a serious challenge is the development of methods that are adaptable to a number of constraints that limit the solution procedure, as for example, resource availability constraints. For this reason, it is important to have a method that constructs document representatives of adjustable size, while preserving important document features.

Consequently, a difficult question emerges, which is: how can we define a 'good' representative for a given full document graph. Having estimated the significance values for all elements of this graph, we can simply apply a filtering procedure on the modeled dataset in order to construct representative graphs. More specifically, we keep the P more important nodes per graph using an evaluation criterion. The evaluation criterion can be based either on the frequency weight of a term resulting in a Frequency Filtering (*FF*), or on the significance weight resulting in the proposed Significance Filtering approach (*SF*). The filtering method is directly associated with the corresponding weighting model, so *SM* uses *SF* and *NWM* uses *FF* respectively.

4 Similarity Measure

4.1 Graph Matching Based on Maximum Common Subgraph

Since we have graph representative models, our sensible aim is to conclude a value $s(G_x, G_y)$ that quantifies the similarity between two given document graphs G_x, G_y. This can be enabled through a *graph matching process* that is based on the maximum common subgraph between the graphs of the corresponding web documents [5, 6]. Even though the *mcs* problem is *NP*-complete in general, in our case we have unique graph labels, therefore we deal a reasonable cost of $O(P)$, where P is the global filtering threshold for all documents. The exact formula is:

$$s(G_x, G_y) = \frac{\left|mcs(G_x, G_y)\right|}{\max(|G_{dx}|, |G_{dy}|)},$$

where $|G|$ is size of graph G (number of graph elements, nodes and edges), $mcs(G_x, G_y)$ is the *mcs* of filtered graphs G_x, G_y and G_{dx}, G_{dy} are the full document graphs (unfiltered). This similarity is called graph-theoretical and is used by *NWM*. Having this idea in mind, we evolve a more appropriate similarity measure.

4.2 Maximum Common Content Similarity Measure

In fact, by finding the *mcs* we simply measure the *size of match*, ignoring whatever information about element significances, even frequencies. We propose the *maximum common content* similarity measure that is based on the significance evaluation of common subgraphs and is used in combination with the *SM*. In particular, we define two elementary similarity cases:

1. $E_w(w_i^{(x)}, w_j^{(y)}) = g_w(w_i, d_x) + g_w(w_j, d_y)$, which measures the similarity that derives from the mutual word $w_i = w_j$, where $w_i \in d_x$ and $w_i \in d_y$
2. $E_e(e_k^{(x)}(w_i, w_p), e_l^{(y)}(w_j, w_q)) = g_e(e_k(w_i, w_p), d_x) + g_e(e_l(w_j, w_q), d_y)$, which measures the similarity that derives from the mutual edge $e_k^{(x)} = e_l^{(y)}$, where $w_i = w_j$, $w_p = w_q$, $e_k \in d_x$ and $e_l \in d_y$.

Supposing that the *mcs* has been calculated, we evaluate the overall normalized similarity by summing on the matched subgraphs:

$$s(G_x, G_y) = \frac{\sum_{i,j,k,l} \left(E_w(w_i^{(x)}, w_j^{(y)}) + E_e(e_k^{(x)}(w_i, w_p), e_l^{(y)}(w_j, w_q)) \right)}{g_D^{(all)}(d_x) + g_D^{(all)}(d_y)} .$$

If we could define the content union of two documents (at the full graph scale), this formula computes a value in the range [0, 1] representing the percentage of common content.

5 Experimental Evaluation

5.1 Clustering Algorithms

We use the typical hierarchical agglomerative algorithm (*HAC*) as a first clustering technique. At first we computer the similarity measure between two clusters c_i, c_j based on the average similarity between cluster elements:

$$sim_1(c_i, c_j) = \frac{1}{|c_i\,||c_j\,|} \cdot \sum_{\forall k,\, d_k \in C_i} \sum_{\forall l,\, d_l \in C_j} s(G_k, G_l),$$

where G_k є c_i και G_l є c_j, $i \neq j$, $k \neq l$.

As we experimentally observed, this measure alone did not provide very good results. For this reason, we considered the following similarity:

$$sim_2(c_i, c_j) \;=\; \min_{k,l} s(G_k, G_l) > 0,\;\; d_k \in c_i \;\; \text{και} \;\; d_l \in c_j .$$

The idea here is that we can measure the correlation of two clusters by finding the minimum similarity between two documents from both sides. In each step, the clusters to be merged would be the two that maximize the similarity measure (*minmax criterion*). This measure, despite of better general performance, has an inherent disadvantage. While clusters are being merged, there is some iteration where the minimum similarity becomes of zero value for all cluster combinations, so as for the max-min similarity. To overcome this difficulty, we introduce a hybrid algorithm that we call HAC_{comb}, which combines both measures as follows: it starts with the second one and when this criterion stops to be useful after a number of iterations, then we change our merging criterion to be the first one (sim_1).

The second algorithm we used is the well-known *k-means* algorithm [10] that has already been used to cluster web documents [2, 5, 7]. Specifically, two versions used: the random center initialization (*RI-KM*) and the *global k-means* (*Global-KM*) [8] which proceeds in an incremental way; it constructs the solution with *k* clusters by optimally adding a new cluster to an already existing solution with *k*-1 clusters. It is deterministic method that is independent of initial conditions. For the k-means type of algorithms we use the median as cluster center, where median m_i is the document that has maximum average similarity with the rest of documents in cluster c_i [9].

5.2 Evaluating Clustering Quality

In our experiments, we evaluate clustering performance using three indices. The first index is the *Rand Index* (*RI*), which is computed by examining all pairs of documents in the dataset after clustering. We consider an *agreement*, if two objects are in the same cluster in both ground truth clustering and final solution. If two objects are not in the same cluster in both ground truth clustering and final solution, this also counts an *agreement*. Any other scenario is considered as *disagreement*. The *RI* is computed by dividing agreements by the sum of agreements and disagreements. This index is a clustering accuracy measure, which is focused on the pairwise correctness of the result.

The second index is a *statistic index* (*SI*), usually mentioned as cluster error, which computes the number of documents being in the “right” cluster after clustering. We can decide about the ‘right’ cluster by finding the dominant class of documents in every cluster, based on ground truth, and thus compute:

$$SI = \frac{1}{N} \cdot \sum_{t=1}^{M} |D_t|,$$

where $|D_t|$ is the number of documents belonging in the dominant class of cluster c_t.

A third index we considered is the *Mean intra-Cluster Error* (*MCE*), which computes the average cluster error, where the error for a cluster is the mean distance between the documents of a cluster and its center. Lower values of *MCE* denote better clustering and better data modeling.

The *RI* and *SI* indices take values in [0,1], where 1 represents the perfect match between ground truth and clustering, while for *MCE*, a lower value indicates better clustering quality.

5.3 Web Document Datasets

For experimental evaluation, we use three web document collections. The *F-series* originally consists of 98 web documents from 4 classes. We altered this dataset, ignoring 5 conflicting multiple classifications. The second dataset is *J-series*, that consists of 185 web documents from 10 classes. We use this dataset unaltered. The reasons for choosing *F-series* and *J-series* are that they provide ground truth assignments, are of moderate size (cluster and document number) and have also been used in other works (available at “ftp://ftp.cs.umn.edu/ dept/users/boley/PDDPdata/”).

We also created an artificial dataset that we call *A-series*. This dataset can help us examine the correctness and appropriateness of all methods using classes of high purity. For the construction of *A-series*, we select sample documents from each class of *F-series*. Then, we select representative phrases from each class without tag information (i.e. plain text). Finally, 10 documents for each of four topics and their ground truth were automatically generated. The content of each document is determined by including a variant number of random phrases and sentences from its ground truth class.

5.4 Experimental Results

We conducted a series of experiments comparing the *NWM* model with the *SM* model proposed in this work. *NWM* uses frequency filtering (*FF*) and assigns no graph weights. The introduced novel *SM* model, on the other hand, uses term filtering based on significance (*SF*) and assigns significance-based weights to graph elements.

Experiments on the A-series dataset. As it can be observed from Table 1, all algorithms and models accomplish satisfactory performance. The fact that documents have only frequency information (no tags) indicates that *SM* results in a *FM* and can help even in case of plain text. No further analysis for the quality of models is possible, because documents exhibit small content intersections.

Table 1. Experimental results on *A-series*

		HAC_{comb}			***RI-KM***			***Global-KM***		
Model	**G/S**	***RI***	***SI***	***MCE***	***RI***	***SI***	***MCE***	***RI***	***SI***	***MCE***
NWM	10	1	1	0.858	0.856	0.785	0.902	0.912	0.89	0.815
	20	1	1	0.641	0.913	0.877	0.703	0.914	0.877	0.631
	30	1	1	0.551	0.92	0.877	0.625	0.965	0.95	0.511
	70	1	1	0.538	0.982	0.975	0.555	0.927	0.9	0.53
SM	10	1	1	0.787	1	1	0.858	0.975	0.975	0.789
	20	1	1	0.553	1	1	0.641	1	1	0.553
	30	1	1	0.471	1	1	0.551	1	1	0.471
	70	1	1	0.448	1	1	0.538	1	1	0.448

Experiments on the F-series dataset. This is a rather "difficult" collection, in which HAC_{comb} fails, since it creates large clusters. However, this problem is less severe when *SM* is used or/and when bigger document graphs are used (*G*/*S*). The results in Table 2 indicate that k-means provides satisfactory solutions and moreover *Global-KM*, as expected, gives much better results than *RI-KM*.

All indices are greatly improved when using *SM* instead of *NWM* for all cases, a fact which indicates that *SM* works better in the case of real documents. Reliable conclusions can be drawn mainly from the *Global-KM* results, since it provides steadily reliable solutions.

Table 2. Experimental results on *F-series*

		HAC_{comb}			***RI-KM***			***Global-KM***		
Model	**G/S**	***RI***	***SI***	***MCE***	***RI***	***SI***	***MCE***	***RI***	***SI***	***MCE***
NWM	10	0.322	0.333	0.999	0.682	0.588	0.998	0.691	0.645	0.998
	20	0.364	0.387	0.998	0.688	0.539	0.997	0.731	0.580	0.996
	30	0.786	0.634	0.994	0.689	0.546	0.995	0.726	0.655	0.994
	70	0.645	0.602	0.986	0.684	0.54	0.987	0.744	0.645	0.983
SM	10	0.622	0.569	0.984	0.699	0.626	0.981	0.736	0.623	0.978
	20	0.619	0.569	0.977	0.702	0.615	0.973	0.807	0.774	0.968
	30	0.628	0.548	0.970	0.708	0.626	0.966	0.738	0.655	0.963
	70	0.779	0.688	0.935	0.718	0.606	0.943	0.785	0.698	0.933

This is the extended version of the work that was published as short paper (4 pages) in the Proceedings of SETN 2006, LNAI 3955, Springer-Verlag, pp. 516-519, 2006.

Experiments on the J-series dataset. This dataset is slightly easier than F-series, which is confirmed by the higher accuracy of the solutions. It contains more and bigger documents, while classes do not have much content intersections. It can be observed in Table 3 that HAC_{comb} stands well on this dataset, and using *SM* gives comparative results to *Global-KM*. An interesting observation is that HAC_{comb} and *Global-KM* have higher quality difference when using *NWM*.

From the experiments using the three datasets, it can be confirmed that the proposed *SM* model for Web document representation along with the *Global-KM* algorithm for clustering, constitute the most powerful combination providing the best results in all cases.

Table 3. Experimental results on *J-series*

		HAC_{comb}			***RI-KM***			***Global-KM***		
Model	**G/S**	***RI***	***SI***	***MCE***	***RI***	***SI***	***MCE***	***RI***	***SI***	***MCE***
NWM	10	0.814	0.529	0.998	0.858	0.530	0.997	0.893	0.621	0.997
	20	0.672	0.389	0.997	0.857	0.548	0.996	0.909	0.697	0.994
	30	0.834	0.594	0.993	0.846	0.507	0.994	0.887	0.643	0.992
	70	0.884	0.637	0.981	0.857	0.530	0.987	0.854	0.545	0.980
SM	10	0.896	0.670	0.934	0.893	0.663	0.939	0.931	0.789	0.916
	20	0.898	0.724	0.929	0.893	0.668	0.918	0.933	0.756	0.900
	30	0.943	0.816	0.898	0.907	0.713	0.918	0.941	0.805	0.893
	70	0.943	0.827	0.878	0.895	0.671	0.902	0.937	0.805	0.869

6 Conclusions

In this paper, we presented a novel significance-based graph model for clustering web documents that assigns significance weights to graph elements (both nodes and edges). An associated similarity measure was also defined, that evaluates the content intersection between two documents using their maximum common subgraph. We evaluated its appropriateness and solution quality for web document clustering by comparing to a pre-existing non-weighted model.

We performed a series of experiments on two real web document collections and an artificial one. In the experimental procedure we used a hybrid hierarchic agglomerative algorithm that we call HAC_{comb} and two alternatives of the widely used k-means, *RI-KM* and *Global-KM*. We considered three evaluation indices (*RI*, *SI*, *MCE*) measuring clustering quality. Experimental results indicate the effectiveness of the proposed *SM* approach. According to results, *SM* is superior to *NWM* in all cases since a clear improvement for all indices was observed in almost all experiments.

In what concerns the clustering algorithms, the agglomerative HAC_{comb} shows some instability on “difficult” data, while when used with the *SM* model, it can be competitive to k-means type of algorithms. From the k-means class of methods, *Global-KM* show a clear qualitative superiority comparing to *RI-KM*, which nevertheless also remains a reliable and computationally ‘cheap’ approach.

This is the extended version of the work that was published as short paper (4 pages) in the Proceedings of SETN 2006, LNAI 3955, Springer-Verlag, pp. 516-519, 2006.

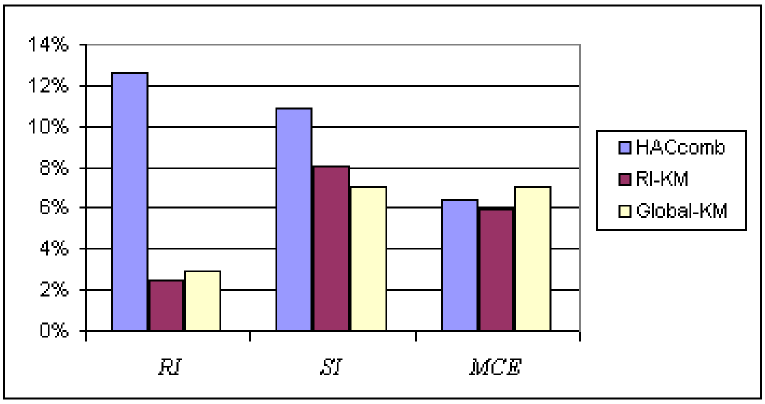

Fig. 2. *SM* vs *NWM* overall improvement measured on all collections using three indices

References

1. R. Kosala and H. Blockeel. Web mining research: a survey. ACM SIGKDD Explorations Newsletter, 2(1):1–15, 2000.
2. A. Schenker, M. Last, H. Bunke and A. Kandel: Clustering of Web Documents Using a Graph Model, Web Document Analysis: Chalenges and Opportunities, eds. A. Antonacopoulos and J. Hu, to appear
3. K. M. Hammuda: Efficient Phrase-Based Document Indexing for Web-Document Clustering, IEEE, 2003
4 M.F. Porter. An algorithm for suffix stripping. Program, 14(3):130–137, July 1980
5. A. Strehl, J. Ghosh and R. Mooney: Impact of Similarity Measures on Web-page Clustering, AAAI-2000: Workshop of Artificial Intelligence for Web Search
6. H. Bunke and K. Shearer: A graph distance metric based on the maximal common subgraph, Pattern Recognition Letters, Vol. 19, 1998, pp. 255–259
7. A.Schenker, M.Last, H. Bunke, A.Kandel: A Comparison of Two Novel Algorithms for Clustering Web Documents, 2nd Int. Workshop of Web Document Analysis, WDA 2003, Edinburgh, UK, August 2003.
8. A. Likas, N. Vlassis and J. J. Verbeek: The global k-means clustering algorithm, Pattern Recognition, Vol. 36, 2003, pp. 451 – 461
9. X.Jiang, A. Muenger, and H.Bunke: On median graphs: properties, algorithms, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence
10. T. M. Mitchell: Machine Learning, McGraw-Hill International Editions, 1997

This is the extended version of the work that was published as short paper (4 pages) in the Proceedings of SETN 2006, LNAI 3955, Springer-Verlag, pp. 516-519, 2006.