# Clustering Web Documents Using Co-citation, Coupling, Incoming, and Outgoing Hyperlinks: a Comparative Performance Analysis of Algorithms

DERRY TANTI WIJAYA
Address required please
*Email: derry.wijaya@gmail.com*

STÉPHANE BRESSAN
*National University of Singapore, Singapore*
*Email: steph@nus.edu.sg*

*Received: April 17 2006; revised: June 12 2006*

***Abstract*— Querying search engines with the keyword "jaguars" returns results as diverse as web sites about cars, computer games, attack planes, American football, and animals. More and more search engines offer options to organize query results by categories or, given a document, to return a list of links to topically related documents. While information retrieval traditionally defines similarity of documents in terms of contents, it seems natural to expect that the very structure of the Web carries important information about the topical similarity of documents. Here we study the role of a matrix constructed from weighted co-citations (documents referenced by the same document), weighted couplings (documents referencing the same document), incoming, and outgoing links for the clustering of documents on the Web. We present and discuss three methods of clustering based on this matrix construction using three clustering algorithms, K-means, Markov and Maximum Spanning Tree, respectively. Our main contribution is a clustering technique based on the Maximum Spanning Tree technique and an evaluation of its effectiveness comparatively to the two most robust alternatives: K-means and Markov clustering.**

***Index Terms*— Search engines, clustering, co-citation, coupling, hyperlinks**

## I. INTRODUCTION

A search in Yahoo [12] for the keyword "fast food" returns links to documents on fast food restaurants, nutrition, and toys. A web user who is interested in finding links to fast food restaurants may have to browse through several result documents before finding links of interest. A solution to this problem is to require the web user to formulate a more precise query, for instance a query that consists of more specific terms. Unfortunately, such a query may in turn returns too few or no result at all. It is indeed difficult for a web user to formulate a precise yet not too restrictive query.

A way to circumvent this problem is for the search engine to offer tools for the grouping of query results into categories or for the access to topically 'related' documents. Instead of browsing through hundreds of more and less relevant results, the user need only find a category of interest or find one document of interest and access a collection of related documents. Some search engines propose such features. Yahoo [12] and DMOZ open directory project [21] organize results according to a manually constructed universal hierarchy of categories. Northern Light [23] uses the text con-tent to cluster results in groups of related documents. The Netscape browser's [24] "what's related" and Google's "similar pages" [22] offer options to retrieve additional documents related to a given page.

While information retrieval traditionally defines similarity between web documents and other documents based on their contents, it seems that the very structure of the web itself carries important information regarding the topical similarity of web documents.

In this paper, following previous work in bibliometrics, we propose to exploit the hypertext structure of the World Wide Web to define notions of weighted co-citation and coupling based on incoming and outgoing hyperlinks from and to common documents. We propose exploit this weighted co-citation and coupling information to define measures of similarity. This idea is based on the simple assumption that two Web documents are likely to be topically related if they either refer to (link to) or they are referred (linked by) by similar documents. We propose a data structure, graph or matrix, to record the link weighted co-citation and coupling information. We present and discuss three methods using this graph or matrix for the clustering of Web documents using three clustering algorithms, K-means, Markov and Maximum Spanning Tree, respectively. The latter algorithm is an original proposal.

In Section II we give an overview of the related work in Web information retrieval and in bibliometrics. We present and discuss the matrix, its construction, and the three clustering algorithms in Section III. In SectionIV we evaluate and comparatively analyze the performance of the three algorithms

using a snapshot of the public Web. We present our conclusions and outline our future work in Section V.

## II. Literature Review

Clustering of Web documents addresses the issue of partitioning a set of Web documents into sub sets of documents that are related. Relatedness is a qualitative and subjective notion to be evaluated by the user or in reference to an existing and accepted classification. To define and quantify relatedness several models of similarity and similarity measures have been defined. Traditionally, information retrieval de-fines similarity in terms of the content of the document. The most commonly used method to measure similarity between documents uses the Vector Space Model [1] for documents. Each document is viewed as a normalized vector of term frequencies. The similarity between two documents is then the inner (dot) product of the vectors which is the cosine of the angle of the vectors in the corresponding multi-dimensional space. The smaller the cosine, the smaller the angle, the more similar are the two documents.

This approach directly applies to Web documents [2]. It is used by most keyword based search engines. However, as such, it works poorly for multimedia documents and document with poor textual content. Furthermore it fails to leverage an important aspect of the World Wide Web, namely its hypertext structure.

In the early 1960s research in bibliometrics lead to the definition of similarity measures based on observable relationship between documents such as co-authorship and bibliographical citations. In [4] Kessler suggested a technique known as bibliographic coupling to measure the similarity between pairs of scientific documents in terms of the number of citations they make in common. In [5] Small proposes another measure called co-citation. Co-citation defines similarity between two scientific documents as the number of common documents that cite both documents. Small and Griffith [6] find clusters of documents using breadth-first-search to compute the connected component of an undirected graph in which vertices are documents and edges represent a positive co-citation value. Persson in [9] studies the comparative proper-ties of coupling and co-citations. He shows that clustering based on coupling documents creates a research front while clustering creates an intellectual base.

Clearly, co-citation and coupling can be transposed to the hypertext structure of the Web where citations are replaced by hyperlinks. For instance Larson [7] proposes an exploratory analysis of co-citation similarity measurement between web documents. Pitkow and Pirolli [8] apply and evaluate the technique proposed by Small and Griffith in [6] to cluster Web documents. They [6] cluster all documents that have at least one document of the co-citation pair in common with the other element in the clusters; without considering the weight of the co-citation. However, they find that this method results in a significant proportion of documents belonging to a few large clusters that consist of diverse set of pages. They [6] then propose another algorithm, based on hierarchical clustering, which considers the Euclidian distance of each co-citation pair. This algorithm performs better and by observation produces well formed clusters. Unfortunately, the details of the algorithm are not discussed and the "goodness" measure of the clusters produced is not conducted. Wang and Kitsuregawa in [11] use the sum of link cocitation and coupling to cluster the documents using an extended K-Means algorithm. They find that the algorithm produces reasonable clusters. Unfortunately the detailed analysis of their experiment results is not presented.

From [6] and [11], one is inclined to believe that the implicit topological structure of hypertext documents namely the co-citation weight and the sum of co-citation and coupling, has the potential to tell us something about the semantic structure of a collection of hypertext documents. In this paper, we would like to explore this potential further by comprehensively comparing the clustering performance when implicit (co-citation weight, coupling weight, sum of co-citation and coupling weights) and explicit (incoming links, outgoing links) information of the topological structure of hypertext documents are each used as a similarity measure to perform clustering using three clustering algorithms: K-means (using a range of k values), Markov clustering and Maximum Spanning Tree clustering - both of which, to our knowledge, have never been applied to the context of the Web. We also propose to employ a notion of weighted co-citation and coupling instead of the usual notion of co-citation and coupling.

## III. Method

### *A. Constructing a Link Graph of Web Documents Based on the Normalized Co-citation and Coupling Similarity Measurements*

We build a weighted graph of link co-citations and coupling. In [14] Ding et al., studying the link structure of the Web, suggest that link co-reference and coupling should be prorated. Their observation equally applies to biometrics and hypertexts. It helps differentiate for instance, co-citation and coupling for survey papers or Web directories, which cite or link many documents, from co-citation and coupling from focused papers and Web documents.

The co-citation value between two documents A and B is normally defined as the number of documents linking to both A and B. The coupling value between two documents A and B is normally defined as the number of documents linked from both A and B. Following the observation of [14] we weight the definition of cocitation and coupling by considering the other links from the referencing and to the referenced documents, respectively.

Let us consider two web documents C, D with links to both documents A and B. Assume that C only links to A and B while D also links to other web documents, say E and F. We define weighted co-citation using the number of outgoing links of C and D. In the example the value of the weighted co-citation between A and B is 2/2 + 2/4 = 1.5.

Similarly, Let us consider two web documents A and B with links to both documents C and D. Let us assume that C is also referenced by other documents E and F. We define weighted

coupling using the number of incoming links to C and D. In the example the value of weighted coupling between A and B is 2/4 + 2/2 = 1.5.

Finally we sum the values of the weighted co-citation and coupling to form the weight of the edge between two documents graph. In practice, for n documents we construct the incidence n × n similarity matrix A, in which the entry aAB corresponding to documents A and B is the sum of the weighted co-citation and coupling values.

Aside from constructing this incidence similarity matrix, for each collection of documents we also construct four other similarity matrices: weighted co-citation matrix, weighted coupling matrix, the inlinks matrix I (in which IAB = 1 if there is an in-coming link to A from B and 0 otherwise) and outlinks matrix O (where OAB =1 if there is an outgoing link from A to B and 0 otherwise). For the rest of this paper, when we mention co-citation and coupling, we mean to refer to the weighted co-citation and weighted coupling.

### *B. Constructing a Link Graph of Web Documents Based on the Normalized Co-citation and Coupling Similarity Measurements*

We can now proceed to clustering the documents using the graph constructed as described above. The purpose of this study is to compare the relative effectiveness of candidate clustering algorithms using various similarity measures. We study three algorithms: the traditional K-means clustering algorithm, the Markov clustering algorithm (MCL), and the Maximum Spanning Tree (MST) clustering algorithm.

**K-Means Clustering**. K-means clustering [25] computes a predefined number k of disjoint clusters. Although K-means clustering is an unsupervised method, the value of k has to be decided beforehand. K-means algorithm starts with an initial random partition of the vertices into k clusters. Subsequent steps iteratively modify the partition to minimize some metrics by moving vertices from one cluster to another. In this paper, the metrics that we use is the sum of cuts (edges weights that straddles the k partitions). We use the METIS graph partitioning library [15]. This package uses heavy-edge matching to firstly coarsen the graph (reduce the size of the graph by collapsing nodes and edges to make the partitioning problem easier), applies a greedy graph growing partitioner (a partitioning algorithm that is similar to K-means but one that tries to minimize the connectivity between partitions) to partition the coarsest graph, and then uses a modified Kernighan-Lin algorithm to un-coarsen the graph back to its original form [16].

**Markov Clustering**. Unlike K-means clustering, Markov clustering [17] does not require the user to supply the number of clusters. Markov Cluster Algorithm utilizes the notion of random walks for the identification of clusters [18]. For a large collection of random walks, all starting from the same point, the walks will in general follow different paths since the direction to be followed at each junction (node) in the graph is chosen randomly according to the weight of the edges. The aim of MCL is to partition the graph into clusters such that, once inside a cluster, the probability for a random walker to leave the cluster is low. MCL iteratively simulates many random walks (also called flow) within the whole graph and strengthen flow where it is already strong and weaken it where it is weak until the graph is partitioned into clusters with strong internal flow separated by boundaries with hardly any flow. We use the Markov Clustering (MCL) package [19]. MCL however requires a parameter to be set (in our package in a range from 1.2 to 5.0). This parameter is called the roughness variable. Higher values yield finer clustering. The complexity of the algorithm, O (n3) where n is the number of nodes (documents) in the graph, mainly lies on the matrix computation it uses to simulate the random walk through the graph represented by the matrix. The nature of the algorithm is similar to finding the fixed point computation of the similarity matrix between documents that are represented as nodes in the graph. Markov clustering, to our knowledge, has never been used in the context of the Web.

**Maximum Spanning Tree Clustering**. A maximum spanning tree of a graph [26] is a tree covering all the vertices of the graph with maximum edges' weights from the original graph. Kruskal algorithm incrementally computes a maximum spanning tree. The algorithm starts with the formation and growth of maximum spanning trees of sub graphs of the initial graph and then merges these sub trees into the maximum spanning tree of the graph. Our hypothesis is that these smaller trees correspond to clusters. Therefore we propose an original clustering algorithm based on a modified version of the algorithm that only computes the component spanning trees. Similarly to Markov Clustering this technique does not require the user to supply the number of clusters. It does not require the setting of any parameter. This idea of using an MST-based algorithm as the basis for clustering was used by Xu et al. [20] for an application in bioinformatics. The complexity of the algorithm, O (n2) where n is the number of nodes (documents), lies mainly on the sorting of edges in the graph (a maximum of n2 number of edges). Maximum Spanning Tree clustering has, to our knowledge, never been used in the context of the Web.

## IV. Experiment and Evaluation

### *A. Building a Snapshot of the Public Web (the Corpus)*

We use the Web, search engines, and Web directories to build an annotated corpus that can let us quantify the effectiveness of the different algorithms.

We first submit queries to Yahoo. We use each of the queries terms: "jaguar", "pyramid" and "fast food". Yahoo's answer contains a list of categories of the Yahoo directory. We manually select some of these categories and the documents they contain. For example, for the query term "jaguar", we select the categories: "Wild Cats ¿ Jaguars", "Automotive ¿ Jaguars", "Video Games Systems ¿ Atari ¿ Jaguar", "Attack Planes ¿ Jaguar" and "NFL ¿ Jacksonville Jaguars". We then expand this set by adding to each document in the set, the first five of its similar documents as returned by Google's 'Similar Documents' feature. This increases the number of relevant documents for different categories, resulting in richer and denser communities of documents. We also manually check

whether each document in the set is reasonably relevant to the query word, and discard irrelevant documents. We refer to the set of documents obtained as the root set. Notice that documents in the root set are labeled with the Yahoo category from which they were obtained directly or using Google's 'similar documents'.

We then expand this root set to form the base set by adding any document pointed to by a document in the root set as well as any document that points to a document in the root set (with at most 10 different documents from different hosts pointing to the document in order to control the size of the base set). The base set contains the root set. The documents in the base set other than those in the root set are not labelled. However, they provide the necessary connectivity to find the different clusters of web documents.

### *B. Evaluation and Metrics*

We build the graph and its similarity matrices for all documents in the base set as explained in 3.1 and apply the three different clustering algorithms: K-means, Markov and MST to find the clusters. Considering only the documents in the root set, which are labelled with categories, we compare the clusters produced with the initial categories. In order to evaluate the performance of a clustering method, for each query word we first compute the precision, recall, and F-values of the clusters produced with respect to each initial category. Table I records the number of root set documents in each cluster that be-long to each of the initial Yahoo's categories. The clusters are computed by MST clustering for the query word "pyramid". There are 8 clusters. 17 out of 90 root set documents labeled in categories pertaining to the query "pyramid" are not grouped and hence are outliers. For each cluster produced, precision, recall and F-values are computed with respect to each initial category.

We need to make a decision with regard to which of the categories the cluster is representing. For this we assign a cluster to the category for which its F-value is the highest. We define the precision ($0 \leq$ Precision $\leq 1$) of a cluster C with respect to category Ct as the number of documents in C belonging to Ct divided by the number of documents in C. Precision represents how pure a cluster (i.e. how many documents in the cluster belong to the same category).

We define the Recall ($0 \leq$ Recall $\leq 1$) of a cluster C with respect to category Ct as the number of documents in C belonging to Ct divided by the number of documents in Ct. Recall represents how relevant a cluster to the category it represents (i.e. how many documents in the category are found in the cluster).

We define the F-value ($0 \leq$ F-value $\leq 0.5$) of a cluster C with respect to category Ct as (precision* recall) / (precision + recall). We assign the cluster to the category for which its F-value is highest. This F-value will be the F-value for the cluster. The cluster's precision is the precision value corresponding to this F-value while the recall is the recall value corresponding to this F-value.

For example, for Cluster 1 in Table I:
The precision with respect to category A: 9/ 9 = 1

TABLE I
NUMBER OF DOCUMENTS PER MST CLUSTER AND CATEGORY
(CATEGORIES OF QUERY WORD "PYRAMID")

| MST Clustering | Category A: Egyptian Pyramids (30 documents) | Category B: Pyramid and Ponzi Schemes (31 documents) | Category C: Food Guide Pyramid (29 documents) |
|---|---|---|---|
| Outliers | 9 | 3 | 5 |
| Cluster 1 | 9 | 0 | 0 |
| Cluster 2 | 12 | 0 | 0 |
| Cluster 3 | 0 | 17 | 0 |
| Cluster 4 | 0 | 9 | 0 |
| Cluster 5 | 0 | 0 | 20 |
| Cluster 6 | 0 | 2 | 0 |
| Cluster 7 | 0 | 0 | 2 |
| Cluster 8 | 0 | 0 | 2 |

The precision with respect to category B: 0/ 9 = 0
The precision with respect to category C: 0/ 9 = 0
The recall with respect to category A: 9/ 30 = 0.3
The recall with respect to category B: 0/ 31 = 0
The recall with respect to category C: 0/ 29 = 0
The F-values with respect to category A: (1 * 0.3) / (1 + 0.3) = 0.23
The F-values with respect to category B: 0
The F-values with respect to category C: 0

The highest F-value for cluster 1 is 0.23 for category A. Thus, we assign cluster 1 to category A, which means that documents in cluster 1 are those representing category A: the Egyptian Pyramids. The representative F-value for cluster 1 is 0.23, its precision is 1 and its recall is 0.3.

### *C. Experiment Results and Analysis*

There are four base sets used in the experiment. Three base sets are each built from the root set corresponding to query words "pyramid", "jaguar" and "fast food" respectively; and one (the combined base set) is built from the combined root sets of query words "jaguar", "java", "gates", "windows", "fast food" and "apple".

In the following we report for each method and each base set (i.e. its corresponding categories) the performance measure (F-value) of the method as computed from the average precision and recall of the clusters it produced after they have been assigned to a category. Five different similarity matrices are being used in the experiment: the incidence matrix (sum of weighted co-citation and coupling), co-citation matrix, coupling matrix, the inlinks matrix and outlinks matrix.

From Figure 1, MST performs the best (has high F-values) when it is used with co-citation matrix or inlinks matrix. This is an interesting result because if MST could produce comparably good clusters by using simply the inlinks matrix, there will be no need for additional computation of co-citation to produce comparable result. The F-values achieved on the combined base set is the highest. This, we believe, is because each community in the combined base set is focused (high intra similarity and low inter similarity). Therefore it maybe easier to extract the communities from this set and higher F-value performance is achieved (Figure 1).

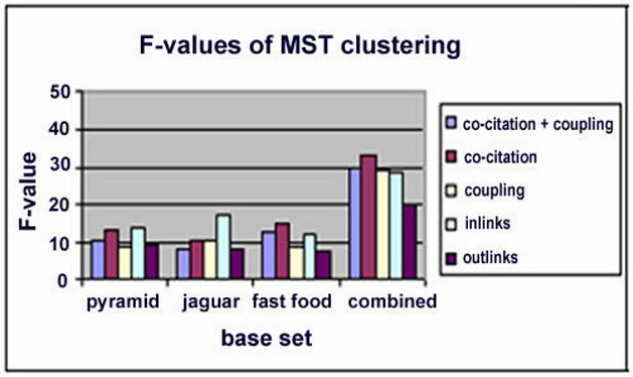



Fig. 1. F-values of MST clustering.

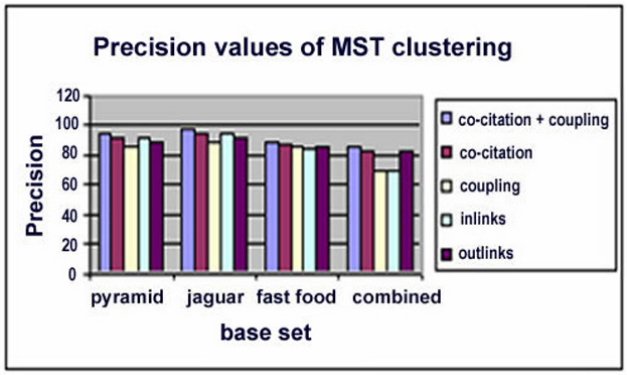



Fig. 2. Precision of MST clustering.

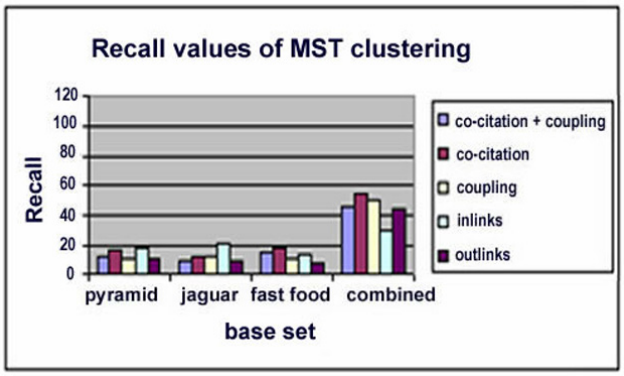



Fig. 3. Recall of MST clustering.

Overall, the F-values of MST clustering are low except on the combined base set. Figures 2 and Figure 3 explain the reason behind this low F-value. Although the average precision of MST clustering is high (Figure 2); its average recall is low (Figure 3). These low recall cause F-values to be low. Further observation into the clusters produced reveals that the low recall but high precision of MST clustering is caused by the existence of some fine but highly precise clusters that the method produces.

One notices on these graphs (Figure 1, 2, 3) that the choice of the F-value as an indicator strikes the best compromised between recall and precision. In the remainder of the study we present results for the best possible F-value of the methods.

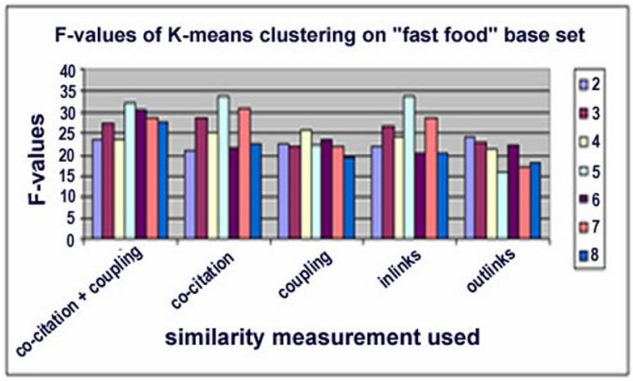



Fig. 4. F-values of K-means on "fast food" data. F-values vary with varying k (k = 2 to 8).

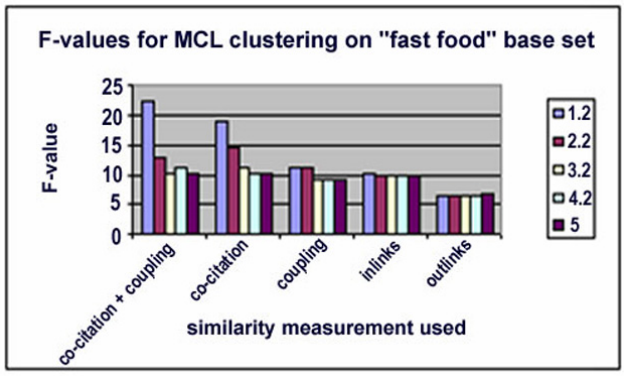



Fig. 5. F-values of MCL on "fast food" data.

The main drawback of the K-means method is that it requires the user to choose a value for k. Similarly, the MCL algorithm is parameterized with a roughness variable. We conduct several experiments as the one reported on Figure 4 and Figure 5 for the categories of the query term "fast food" in order to manually select the best setting for the most competitive comparison.

We find out from Figure 4 that K-means performs best on fast food data (highest F-value) with k = 5, which is the number of initial categories in fast food root set. We see from Figure 5 and Figure 6 that MCL globally performs best for a roughness value of 1.2. In general the performance of K-means is very sensitive to the value of its parameter and performs best when a correct value of k is supplied. This is significantly less the case for MCL. For a roughness value of 2.2, the performance of MCL does not vary much.

From Figure 7, we find that MCL clustering performs best when used with co-citation matrix or incidence matrix (sum of co-citation and coupling).

Using MCL clustering with co-citation matrix produces comparably good clusters as using incidence matrix. This means that there will be no need for additional computation of coupling to produce comparable result.

Similar to MST clustering, the average precision of MCL is high (Figure 8) but its average recall is low (Figure 9). Similar to MST, MCL produces fine but highly precise clusters.

From Figure 10, we find out that K-means clustering per-

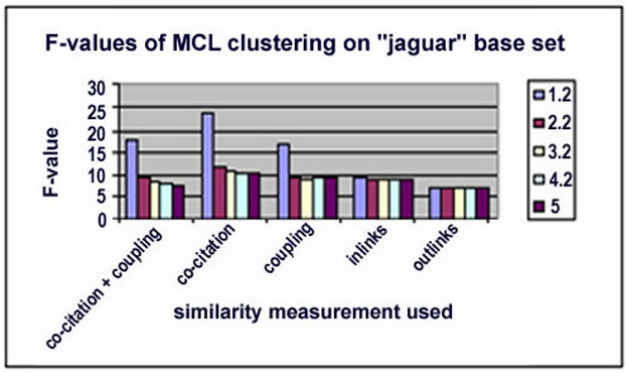



Fig. 6. F-values of MCL on "jaguar" data.

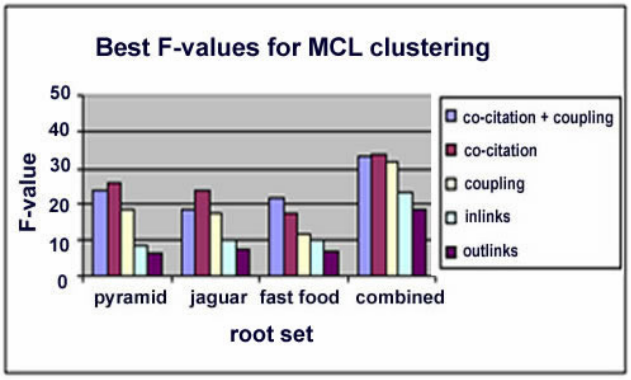



Fig. 7. Best F-values of MCL.

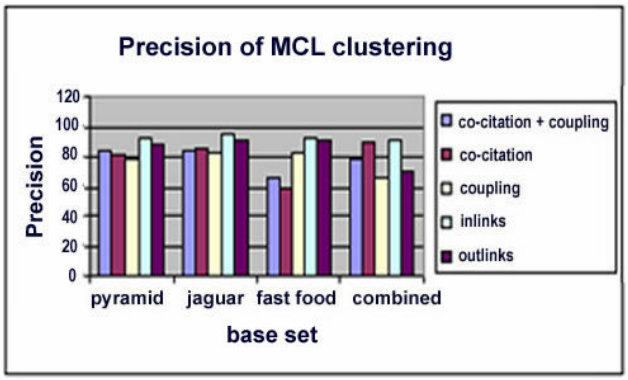



Fig. 8. Precision corresponding to MCL best F-values.

forms comparably when used with the incidence, co-citation, coupling and inlinks similarity matrices. It performs the worst when used with outlinks similarity matrix.

Figure 11, 12 and 13 compare the precision, recall and F-values of MST, MCL and K-means clustering on fast food data.

From Figure 11 and 12, we can see that MST and MCL outperform K-means in terms of precision, but are outperformed by K-means in terms of recall. This is because MST and MCL produce some fine clusters that result in low average recall. Due to this low recall, the F-values of MST and MCL are outperformed by the F-values of K-means (Figure 13).

Since MST and MCL produce some fine but highly precise

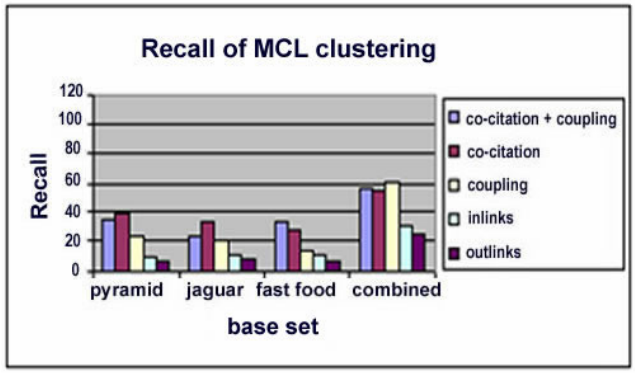



Fig. 9. Recall corresponding to MCL best F-values.

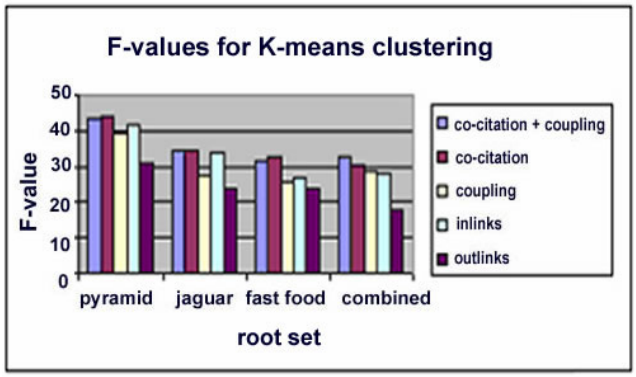



Fig. 10. Best F-values of K-means clustering.

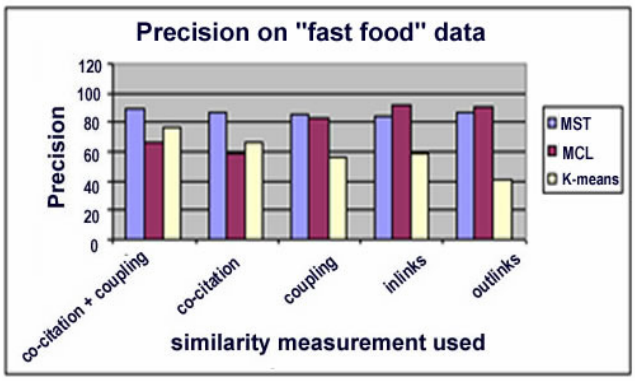



Fig. 11. Precision on fast food data. MST and MCL outperforms K-means in terms of precision.

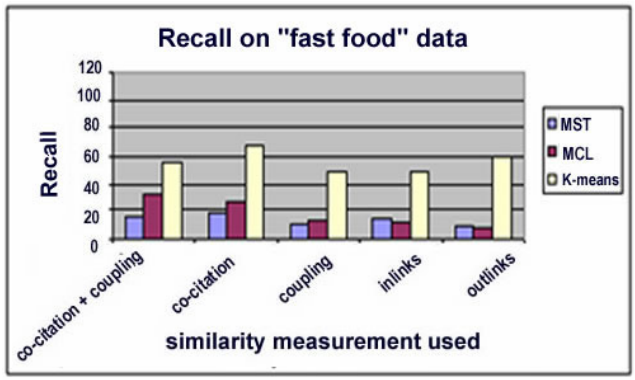



Fig. 12. Recall on fast food data. K-means outperforms MST and MCL in terms of recall.

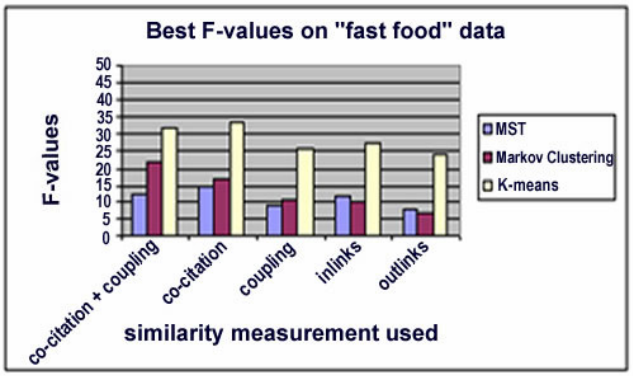



Fig. 13. F-values on fast food data under the best possible settings.

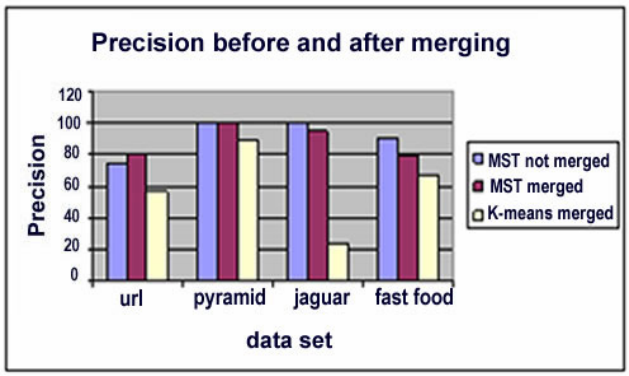



Fig. 14. Precision of MST and K-means using inlinks similarity matrix.

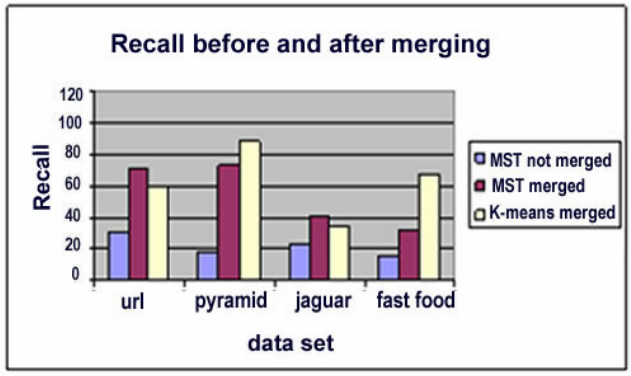
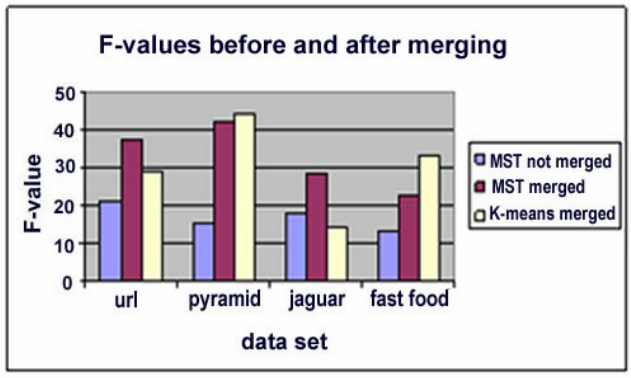



Fig. 15. Recall of MST and K-means using inlinks similarity matrix.




¿

Fig. 16. F-values of MST and K-means using inlinks similarity matrix.

clusters, we believe that further merging of these clusters will improve the average recall and ultimately F-values. We conduct further merging of MST clusters whose total cut (sum of edges in between them) is X% of the total connectivity of their clusters (sum of edges inside their clusters). We try different values of X and find that the performance (F-value) is highest when X=10. We conduct this merging on clusters produced with inlinks similarity matrix; as MST performs best when used with inlinks similarity matrix.

From Figure 14, 15 and 16, we can see that the average recall of MST improves after merging. Its average precision is not much affected by the merging and its F-values improve. We compare this improvement with the precision, recall and F-values of K-means clusters after we merge (manually) the K-means clusters belonging to the same category together.

We can see from Figure 14 that the precision of MST is much higher than that of K-means. From Figure 15, we can see that the recall of MST improves largely after merging and its F-values become comparable to that of K-means. This shows that a good merging criterion could improve the performance of MST and makes its performance comparable or even higher than that of K-means. This is an advantage as MST, unlike K-means, does not require the user to supply any parameter beforehand and can perform well using simple incoming links analysis (inlinks similarity matrix).

## V. Conclusion

In this paper we have explored the use of weighted co-citation and coupling for the clustering of Web pages. We have empirically compared the three clustering algorithm: K-means, MCL, and MST clustering algorithm on the incidence matrix, co-citation matrix, coupling matrix, inlinks matrix and outlinks matrix of each collection of Web documents. Markov and MST clustering have, to our knowledge, never been used in the context of the Web.

Our results show that MCL and MST yield the best precision. K-means yields the best recall. But K-means is ruled out by the fact that it requires the correct setting of the number of clusters beforehand. At comparable effectiveness, MST is more efficient (less complexity) than MCL.

Our results have also shown that MST performs best when used with co-citation matrix or inlinks matrix. This would mean that comparably good clusters can be produced by MST using simply the inlinks matrix.

On the other hand, MCL performs best when used with co-citation matrix or incidence matrix. This would mean that comparably good clusters can be produced by MCL using only the co-citation matrix.

Since the nature of MCL algorithm equals to finding the fixed point computation of the similarity matrix used, the fact that MCL performs best when used with co-citation matrix corresponds to finding the fixed point computation of the co-citation matrix. As Kleinberg in [3] finds, the fixed point computation of co-citation matrix corresponds to finding the

authoritative documents in hyperlinked environment. This will mean that using MCL with co-citation matrix to produce clusters can, at the same time, find the authoritative documents in the collection. Furthermore, the clusters produced using this method may actually correspond to the "non-principal eigenvectors" that represent different communities in a collection of hyperlinked documents [10].

Furthermore, when merging of clusters is conducted, the F-values of MST and MCL are seen to improve and be comparable or bigger than that of K-means clustering. This calls for a good merging criterion to be defined to merge the highly precise (but low-recall) MST or MCL clusters to improve their recall and F-values.

In general we now need to conduct a finer grain analysis in order to understand the nature and granularity of clustering as well as (similarly to the work by Persson (1994)) the effect and role of linkages with different nature: such as links in embedded advertisements, structural links to table of contents etc. Furthermore, there is the issue of naming the clusters produced. Automatically obtaining context-relevant tags for each cluster, possibly by using the traditional information retrieval technique on the contents of documents or their snippets, will be the next issue to discuss.

Although we have been able to characterize the detailed performance of the three algorithms presented and studied, these algorithms and the data structure they exploit work well for the offline clustering of hypertext collections. Additionally, although the proposed techniques can process large corpora, they may not reasonably scale to the billions of documents on the public Web. As such they apply to defined collections such as digital libraries, corporate web sites, or to support decision on perennial components of the Web. In order to work for generic search they need to be adapted to process larger and more dynamic collections in which documents are continuously created, changed and deleted. To do so, we now need to devise a data structure that can be maintained incrementally and variants of the algorithms that can work online. It is likely that this can only be obtained at the expense of exactitude. However our MST clustering algorithm, thanks to its simplicity, has the highest potential to develop into an effective and efficient online version especially if it can find good clustering by using only the simple incoming links analysis (inlinks matrix) of the documents.

## REFERENCES


[1] Salton, G. Automatic Text Processing. Addison Wesley, Massachusetts (1989)

[2] Yuwono, B., Lam, Savio L., Ying, Jerry H., Lee, Dik L. A World Wide Web Resource Discovery System. *The Fourth International WWW Conference*, Boston, USA, December 11–14 (1995)

[3] Kleinberg, Jon M. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, Volume 46, Issue 5, 604–632 (1999)

[4] Kessler, M. M. Bibliographic Coupling between Scientific Papers. *American Documentation*, 14, 10 (1963)

[5] Small, H. G. Co-citation in the Scientific Literature; a New Measure of the Relationship between Two Documents. *Journal of the American Society for Information Science* 24: 265–269 (1973)

[6] Small, H., Griffith, B. C. The Structure of the Scientific Literatures I: Identifying and Graphing Specialties. *Science Studies* 4, 17–40 (1974)

[7] Larson, R. R. Bibliometrics of the World Wide Web: an Exploratory Analysis of the Intellectual Structures of Cyberspace. *Proc. SIGIR 1996*, documents 71–78 (1996)

[8] Pitkow J., Pirolli, P. Life, Death, and Lawfulness on the Electronic Frontier. *Proc ACM SIGCHI* (1997)

[9] Persson, O. The Intellectual Base and Research Front of JASIS 1986–1990. *Journal of the American Society for Information Science* 45 (1), pp. 31–38 (1994)

[10] Gibson, D., Kleinberg, J., Raghavan, P. Inferring Web communities from link topology. *Proc. 9th ACM Conference on Hypertext and Hypermedia* (1998)

[11] Wang Y., Kitsuregawa, M. Link Based Clustering of Web Search Results. *WAIM 2001*, 225–236 (2001)

[12] Yahoo Search Engine: http://ww.yahoo.com

[13] Google Search Engine: http://www.google.com

[14] Ding, C., He, X., Husbands, P., Zha, H., Simon, H. DocumentRank, HITS and a Unified Framework for Link Analysis. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, documents 353–354 (2002)

[15] Karypis, G. METIS Family of Multilevel Partitioning Algorithms http://www-users.cs.umn.edu/karypis/metis/

[16] Alpert C. J., Hagen L., Kahng, A. B. A Hybrid Multilevel/Genetic Approach for Circuit Partitioning. *Proc. ACM SIGDA Physical Design Workshop*, April 1996, pp. 100–105 (1996)

[17] van Dongen, S. Graph Clustering by Flow Simulation. PHD thesis, University of Utrecht (2000)

[18] Nieland, H. Fast Graph Clustering Algorithm by Flow Simulation. *ERCIM News*, No. 42, July 2000 (2000)

[19] van Dongen, S. Markov Clustering Package. http://micans.org/mcl/

[20] Xu, Y., Olman V., Xu, D. Minimum Spanning Trees for Gene Expression Data Clustering. *Proceedings of the 12th International Conference of Genome Informatics (GIW)*

[21] DMOZ open directory project: http://www.dmoz.org

[22] Google's 'Similar Documents' feature: http://www.google.com/help/features.html#related

[23] Northern Light Search Engine: http://www.northernlight.com/

[24] Netscape Browser "What's related" feature: http://wp.netscape.com/escapes/related/

[25] MacQueen, J. Some methods for classification and analysis of multivariate observations. In: Le Cam, L. M. and Neyman, J., editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume I: Statistics, 281–297 (1967). University of California Press, Berkeley and Los Angeles, CA

[26] Gondran, M., Minoux, M. *Graphes et algorithmes*. Eyrolles (1985)



**Derry Tanti Wijaya** is a candidate to the graduate research program of the Computer Science department of the School of Computing (SoC) at the National University of Singapore (NUS). Her research interests include algorithms and their application to Web applications.

**Stéphane Bressan** is senior lecturer in the Computer Science department of the School of Computing (SoC) at the National University of Singapore (NUS) and adjunct associate professor at Malaysia University of Science and Technology (MUST). His research is pertaining to the integration of heterogeneous and distributed information sources and services.