Taking Advantage of Out-of-Corpus Information for Citation Network Clustering

Steven Lee STLEE@CS.UMD.EDU
University of Maryland, College Park, MD 20742 USA

Taesun Moon TSMOON@UMIACS.UMD.EDU
University of Maryland, College Park, MD 20742 USA

Hal Daume III HAL@UMIACS.UMD.EDU
University of Maryland, College Park, MD 20742 USA

Abstract

In this paper we explore the use of several popular clustering and graph partitioning algorithms as a method of generating clusters of related scientific documents and suggest a simple graph augmentation technique for taking advantage of external information. We show that by hallucinating nodes for scientific documents that are cited but not present in the original dataset, we can improve performance of clustering algorithms.

1. Introduction

Clustering is an important unsupervised task for conducting data analysis, dimensionality reduction and pattern extraction among others with many practical applications (Jain et al., 1999; Zamir & Etzioni, 1998; Zeng et al., 2004). One particular form of clustering focuses on citation graphs extracted from scientific corpora or link structures from web corpora (Newman & Girvan, 2004; Fortunato, 2010). In a practical setting, these citations are extracted from a text corpus (either structured or unstructured) to create a directed or undirected graph where documents constitute nodes. Unfortunately, such corpora often contain outgoing links or citations to documents that are not contained in the corpus. As such, graph clustering algorithms naively working with such corpora are based on incomplete data and may arrive at faulty or deficient conclusions (Hopcroft et al., 2004) as we empirically demonstrate in this paper. Furthermore, because the degree distributions in such corpora have power law distributions (Newman & Girvan, 2004), it becomes even more difficult for such algorithms.

Using the open access subset of PubMed [1] and simulated data, we propose a set of graph augmentation techniques to take advantage of this information and thoroughly examine three well-studied clustering algorithms and their performance on naive and augmented graphs. Because gold label data is hard to come by in clustering problems, we use pseudo-labels such as the PubMed MeSH labels and measures of textual cohesion to evaluate performance for PubMed. We use the generated cluster labels for simulated experiments. While results are mixed for PubMed data, we show that these simple graph modifications can provide a significant boost to community detection performance across all algorithms on simulated graphs.

2. Models

Let $G = (V, E, W)$ be a graph where V is the set of nodes, E the set of edges and W the weights over the edges. $n = |V|$ is the number of nodes in the graph, and $W \in \mathbb{R}^{n \times n}$ is defined as a weighted adjacency matrix. We also define an expanded (or *hallucinated*) graph $G_h = (V_h, E_h, W_h)$ such that $V \subset V_h, E \subset E_h$. Then given $m = |V_h|$, $W_h \in \mathbb{R}^{m \times m}$ is the weight matrix for the hallucinated graph.

When extracting the graph from a corpus with link structure, V is the set of documents in the corpus, the set of nodes $V_f = V_h \backslash V$ is defined to be the hallucinated *frontier*, i.e. the set of documents which don't exist in this corpus but are cited by documents/nodes in V. The input graphs derived from text corpora with

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1] ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/

link structure are augmented in two different ways, which are examined separately. The methods are:

- **node hallucination**: A document j that is cited by $i \in V$ but is not in V is added to V_f. A corresponding edge (i, j) is added to E_h and weight matrix W_h.
- **edge hallucination**: If two documents in V cite a common document in V_f, an artificial edge with one half weight is added between the two documents. In matrix notation, we apply clustering algorithms to the matrix $W_h^T W_h$.

Below, we briefly describe the models which form the basis of our experiments: spectral clustering (Ng et al., 2001), Louvain method (Blondel et al., 2008) and Metis (Abou-Rjeili & Karypis, 2006). These popular algorithms vary widely in technique, and the results shown here provide hints as to how other untested algorithms may perform (Schaeffer, 2007; Jain, 2010; Fortunato, 2010).

2.1. Spectral Clustering

Spectral Clustering, which has been show to be effective and reasonably fast, finds an eigendecomposition of a modified version of the graph's original adjacency matrix, and then uses its largest eigenvectors as reduced dimension inputs for k-means clustering (Ng et al., 2001). The graph partitioning found approximates the minimization of the normalized cut score of the graph. In theory it should find partitions with small edge cuts, and similar cluster sizes.

Despite the availability of sparse eigendecomposition algorithms like Arnoldi iteration, Spectral Clustering is the slowest algorithm used here by an order of magnitude.

2.2. Louvain Method

The Louvain Method is a greedy, agglomerative graph clustering algorithm that locally and iteratively maximizes the modularity function: (Blondel et al., 2008)

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j)$$

where A is the adjacency matrix representing the graph, k_i is the degree of node i, m is the sum of all the edge weights in the graph, and c_i is the cluster assignment. $\delta(c_i, c_j) = 1$ if $c_i = c_j$ and 0 otherwise. The range of Q is $[-0.5, 1]$.

Intuitively, the modularity score of a graph is high if in each found cluster, the ratio of edges between nodes within the cluster to the total edges with at least one endpoint in the cluster is greater than the ratio expected if all edges were attached randomly.

2.3. Metis

The graph partition algorithm from the Metis software collection attempts to find partitions with a minimum cut score and works in three stages: a coarsening stage where nodes and edges are iteratively collapsed, a partitioning stage on the coarsened, more tractable graph, and an expansion and refinement stage where the Kernighan-Lin algorithm is run at each step of the expansion (Abou-Rjeili & Karypis, 2006).

This approach has been known to be both extremely effective and fast. While it attempts to minimize cut score, similar to Spectral Clustering, it does so via a completely different method that produces very regular partitions (Abou-Rjeili & Karypis, 2006).

3. Data

To evaluate the effect of graph augmentation on clustering algorithms, we use two types of data. One is real world data from PubMed that lacks gold labels and the other is simulated data with gold labels, described below.

3.1. PubMed collection

The real-world data used in this paper comes from the Open Access subset of the PubMed collection of scientific documents.[2] The collection consists of over 200,000 full text documents from various journals with a bio-medical focus. As in other scientific corpora that we have examined, the difficulty of this data is that the collection is not complete: a vast majority of the citations within the documents in the collection resolve to documents outside of the collection.

We build the citation network from the collection and take the largest connected component (composed of nearly 80,000 documents and 200,000 citations) as our naive graph, and then create the expansions described above by generating the frontier of the network. Basic information on these networks are presented in Table 1, including the average clustering coefficient for each. As can be expected from the edge counts relative to the number of nodes, the average clustering coefficient is very low for the naive graph, a condition that has the potential to make it difficult for graph partitioning algorithms to find meaningful communities within the network. The expanded networks both

[2] ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/

have higher coefficients: slightly higher for the graph with hallucinated nodes, and significantly higher for the graph with hallucinated edges.

Table 1. PubMed Graph Statistics

network	nodes	edges	clustering coeff.
Naive	85465	211036	0.107
Hal. Nodes	554186	2097662	0.122
Hal. Edges	85465	15019995	0.352

3.2. Simulated data

Due to the difficulty of finding gold label clusters for the citation data, we also run simulation experiments in order to further explore the algorithms' behavior on the augmented graphs. To generate gold-label data we create a simple problem. First, four directed subgraphs of 1000 nodes each are made using the Forest Fire method of graph generation, which has been shown to create graphs with properties similar to real world citation networks (Leskovec et al., 2007). The nodes in these disconnected subgraphs are the gold label cluster data. Then, 100 directed edges are randomly added to the graph with the constraint that the endpoints must be from separate subgraphs. To simulate missing data, 20% of the nodes in the entire graph are randomly selected and marked as missing—these missing nodes become the *frontier* V_f—and the naive and augmented graph types with hallucinated nodes and edges are created from this incomplete simulated graph.

For the naive graph, any nodes in the frontier and any edges incident to those nodes are simply deleted. To create the hallucinated nodes graph, edges with their source in the frontier are removed; nodes in the frontier as well as edges with a destination in the frontier but a source node not within the frontier are retained. The hallucinated edge graph is generated by collapsing the frontier nodes into a set of edges between each non-frontier node with a link to the frontier nodes. We generate 1000 of these simulated graphs to get an average of the performance measures described in the next section.

4. Evaluation

We use a set of general metrics as well as data specific metrics on the PubMed data and the simulated data. For both data sets, we use standard evaluation metrics such as precision/recall/f-scores and information theoretic metrics. This is straightforward for the simulated data sets since gold labels are generated with the data. Because no such labels exist for PubMed, we use MeSH categories as pseudo-labels. In addition, full text is available for the PubMed data and so we evaluate the clusterings based on measures of textual cohesion. These measures are applied to the naive as well as the augmented graph inputs.

4.1. PubMed specific experiments and evaluation

First, measures of textual cohesion are applied including Davies-Bouldin (Davies & Bouldin, 1979) and normalized sum of squared error (both calculated using pruned tf-idf vectors generated from the paper abstracts). We make the assumption that if a set of papers make up a scientific community, they will be more similar in text to themselves than papers from other clusters.

Davies-Bouldin is a measure used to determine the quality of a cluster using the inherent qualities of the data, in this case text. It can be loosely described as the ratio of similarity within each cluster to the similarity between each cluster and its closest neighbor cluster. Lower scores are better and represent clusterings that are similar when nodes are compared internally, and dissimilar when nodes are compared to other clusters.

The normalized sum of squared error is a precision based metric that measures the weighted distance of each node to its cluster's centroid. Lower values mean that the clusters are composed internally of more similar nodes.

Note that we do not use text in the clustering phase because we are only attempting to measure the effects of the augmented citation networks. It is often the case that for an initial clustering or analysis, full text similarity may be too slow or impractical; clustering purely on the citation network or graph information of a dataset is an alternative tool that can still generate high quality results while requiring only a fraction of the time necessary for full-text based methods.

Second, we take advantage of the Medical Subject Headings (MeSH) labels (HJ & G, 1994; Ruiz & Srinivasan, 1999) for the documents in the network and calculate normalized mutual information and purity in addition to the standard precision, recall, and f-score measures. For MeSH, each document is labeled with multiple, often hierarchical labels, each representing a general subject discussed in the paper. Precision and recall based statistics are measured by assigning weight of 1 for each MeSH label in the confusion matrix, attributing to the exceptionally low seeming

Table 3. Text Cohesion Measures (Davies-Bouldin and Normalized Sum of Squared Error)

Algorithm, Graph	DB ($\times 10^1$)	NSqErr ($\times 10^{-1}$)
SC, Naive	**0.967**	4.175
SC, Hal. Nodes	1.143	4.175
SC, Hal. Edges	1.054	**4.152**
Louvain, Naive	0.453	1.708
Louvain, Hal. Nodes	**0.432**	**0.663**
Louvain, Hal. Edges	0.837	4.704
Metis, Naive	1.557	4.133
Metis, Hal. Nodes	**1.226**	**4.126**
Metis, Hal. Edges	1.685	**4.126**

scores. As with the textual cohesion measures, here we also make the assumption that the MeSH labels are indicative of scientific community.

4.2. Simulation

We apply the clustering algorithms to the simulated data as described in sec. 3.2. The algorithms are applied to the naive as well as the augmented graphs and the output is evaluated using the gold labels in terms of precision/recall/f-score, purity, and NMI metrics and then averaged over the 1000 generated graphs. Spectral Clustering and Metis are both explicitly set to find four clusters.

5. Results

The three algorithms used in this paper operate very differently, and consequently find very different partitions. The following analyses for each algorithm attempts to give the reader a feel for the type and quality of the partitions found by each algorithm on PubMed and simulated data.

5.1. PubMed results

Table 3 shows the results of the textual cohesion metrics, and Table 2 shows the results of the MeSH evaluation metrics. Note that for computational feasibility, clusters with fewer than three nodes were not included in the text evaluations.

5.1.1. Spectral Clustering

Spectral Clustering, which was run with parameters set to find fifty eigenvalues and one hundred clusters, creates unsatisfying partitions on the naive PubMed graph: about half of the nodes are all put into a single dominating cluster. Despite this partitioning's low cut score, it doesn't seem to find distinct communities. When run on the graphs with hallucinated nodes and edges, the cluster sizes are slightly more equitable, with the largest cluster having only one ninth of the nodes in the total graph. The cluster size distribution can be better seen in the Figures 1 and 2, which show what fraction of the graph nodes are contained in clusters of varying sizes.

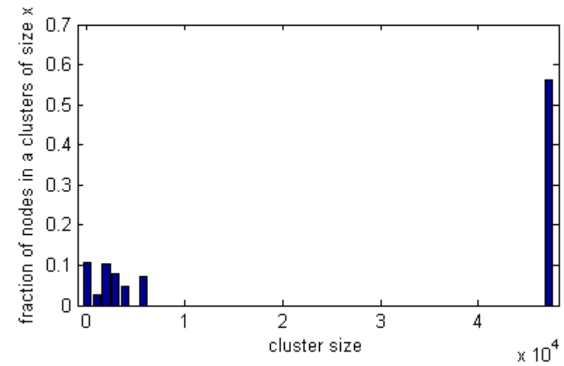

Figure 1. Cluster size distribution for spectral clustering on the naive PubMed graph

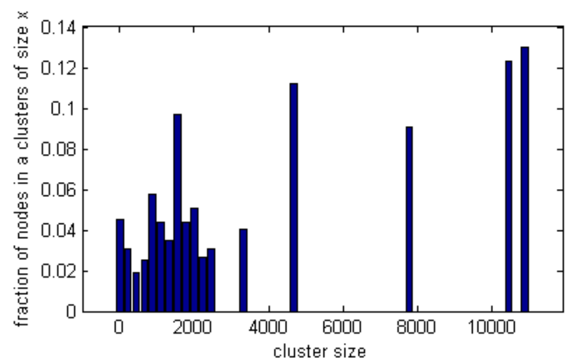

Figure 2. Cluster size distribution for spectral clustering on the PubMed graph with hallucinated nodes

For the text evaluation, Spectral Clustering performs best on the naive graph for Davies-Bouldin and best on the hallucinated edge graph for normalized sum of squared error, although the differences aren't significant. For the MeSH evaluation, Spectral Clustering performs best with the naive graph for precision and recall, but is beaten on normalized mutual information by the graph with hallucinated edges. Figure 3 shows the text normalized squared error as the number of clusters changes, while Davies-Bouldin performance is less clean. We believe that better results will be found if time is unrestrained and experiments using greater numbers of eigenvalues are ran.

Table 2. PubMed MeSH Evaluation Measures

Algorithm, Graph	purity ($\times 10^{-2}$)	NMI ($\times 10^{-1}$)	precision ($\times 10^{-3}$)	recall ($\times 10^{-3}$)	f-score ($\times 10^{-3}$)
SC, Naive	**5.30**	0.205	**6.87**	**3.10**	**4.28**
SC, Hal. Nodes	**5.30**	**0.273**	6.86	0.383	0.726
SC, Hal. Edges	**5.30**	0.245	6.86	0.556	1.03
Louvain, Naive	5.88	1.85	6.75	0.00346	0.00692
Louvain, Hal. Nodes	**7.04**	**2.20**	6.78	0.0367	0.0730
Louvain, Hal. Edges	5.32	0.350	**6.94**	**0.319**	**0.610**
Metis, Naive	**6.56**	**2.46**	6.44	0.152	0.297
Metis, Hal. Nodes	**6.56**	2.44	**7.09**	**0.176**	**0.343**
Metis, Hal. Edges	6.48	2.44	5.91	0.140	0.273

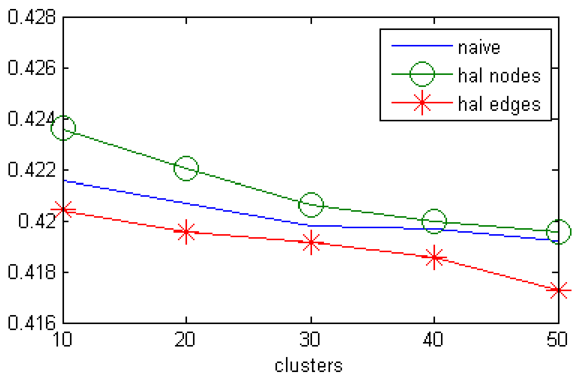

Figure 3. Normalized squared error of textual cohesion in relation to number of clusters for spectral clustering

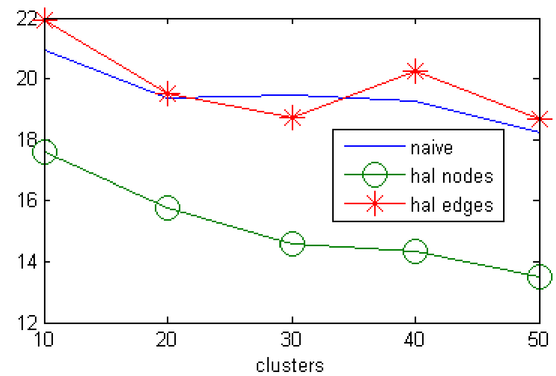

Figure 4. Davies-Bouldin measure of textual cohesion for spectral clustering in relation to number of clusters for Metis

5.1.2. Louvain Method

The clusters found by the Louvain method on the naive and hallucinated node graphs have the opposite flaw as those found Spectral Clustering. Here, almost all of the nodes are placed into tiny clusters with fewer than ten and twenty nodes (for the naive and hallucinated node graphs respectively), which we do not believe to be representative of general communities. When run on the PubMed graph augmented with hallucinated edges, Louvain gives drastically improved results by finding many large clusters with over one thousand nodes each.

Louvain has very low (good) measures for Davies-Bouldin and Normalized Sum of Squared Error when the naive and hallucinated node graph is used, but this is to be expected since these measures are more precision based, and thus the tiny clusters perform very well. For MeSH, the graph with hallucinated edges drastically outperforms the other alternatives due to the larger, more general clusters and therefore significantly improved recall score.

5.1.3. Metis

Metis, which was set to find two hundred clusters, created partitions with unusual regularity. When run on the PubMed graph variations Metis finds a partitioning where all clusters are nearly the exact same size, all varying only by only tens of nodes from the average. While this challenges our presumed intuition about the varying sizes of scientific communities, these partitions prove to be very robust in nearly all evaluation measures.

While the normalized sum of squared error score remains almost completely static, the Davies-Bouldin score is best when the graph with hallucinated nodes is used. Figure 4 show that this trend holds true as the number of clusters found is varied. For MeSH, Metis gives significantly better, precision, recall, and f-scores when run on the graph with hallucinated nodes.

Table 4. Simulation Results

Algorithm, Graph	purity	NMI	precision	recall	f-score
SC, Naive	**0.272** (0.032)	0.00663 (0.013)	0.253 (0.010)	0.773 (0.190)	0.372 (0.036)
SC, Hal. Nodes	0.271 (0.035)	0.00906 (0.020)	0.254 (0.013)	0.847 (0.243)	0.378 (0.042)
SC, Hal. Edges	**0.272** (0.041)	**0.0109** (0.028)	**0.256** (0.019)	**0.867** (0.214)	**0.384** (0.034)
Louvain, Naive	0.958 (0.004)	0.370 (0.011)	0.933 (0.013)	0.0712 (0.027)	0.131 (0.045)
Louvain, Hal. Nodes	0.960 (0.004)	0.382 (0.012)	0.941 (0.010)	0.0908 (0.027)	0.164 (0.045)
Louvain, Hal. Edges	**0.964** (0.004)	**0.427** (0.014)	**0.943** (0.010)	**0.176** (0.038)	**0.295** (0.053)
Metis, Naive	0.862 (0.021)	0.615 (0.039)	0.752 (0.033)	0.752 (0.033)	0.752 (0.033)
Metis, Hal. Nodes	0.924 (0.015)	0.761 (0.036)	0.858 (0.027)	0.858 (0.027)	0.858 (0.027)
Metis, Hal. Edges	**0.933** (0.020)	**0.782** (0.044)	**0.873** (0.034)	**0.873** (0.034)	**0.873** (0.034)

Average over 1000 randomly generated graphs. Standard deviation in parenthesis.

5.2. Simulation results

This section provides results on simulation. Table 4 shows the averaged results for 1000 simulated graphs and evaluations. We go into more detail in the following sections.

5.2.1. Spectral clustering

The simulated graphs, which should be theoretically easy to separate provides trouble for Spectral Clustering. Similar to the PubMed partitions, Spectral Clustering tends to selects one very large and three much smaller clusters on the simulated graphs. In all of our experiments, Spectral Clustering selects heavily unbalanced, trivial partitions despite it's approximate minimization of the normalized cut score.

5.2.2. Louvain method

Louvain behavior on the simulated graphs is consistent with its behavior on the PubMed graphs. While it selects over 400 clusters instead of the one cluster for each of the gold labels, the clusters have very high purity and precision scores. The algorithm chooses larger more general clusters on the graphs augmented with additional edges, and thus has the highest recall and f-scores on the those graphs.

5.2.3. Metis

As can be seen from immediately from the simulated results 4 Metis has very high performance on all three graph variants. The augmented graphs both perform significantly better compared to the naive graph, with the hallucinated edges graph only slightly outperforming the graph with hallucinated nodes.

6. Conclusion

We have shown that graph augmentation using out-of-corpus information has the potential to enhance performance of partitioning algorithms for use in community detection when applied to citation networks. The results are mixed for the real world data of PubMed, where Louvain and Metis benefit from having augmented graphs as input but Spectral Clustering does not. On the other hand, it is clear that graph augmentation can provide significant gains in performance to standard clustering algorithms on simulated data designed to mimic the scientific publication and citation process. Furthermore, we have discovered severe gaps in performance between clustering algorithms for this particular type of simulated data. We hope to investigate this phenomenon further in future work.

References

Abou-Rjeili, A. and Karypis, G. Multilevel algorithms for partitioning power-law graphs. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pp. 10 pp., april 2006. doi: 10.1109/IPDPS.2006.1639360.

Blondel, Vincent D, Guillaume, Jean-Loup, Lambiotte, Renaud, and Lefebvre, Etienne. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008. URL http://stacks.iop.org/1742-5468/2008/i=10/a=P10008.

Davies, David L. and Bouldin, Donald W. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):224 –227, april 1979. ISSN 0162-8828. doi: 10.1109/TPAMI.1979.4766909.

Fortunato, S. Community detection in graphs. *ArXiv*, 486:75–174, February 2010. doi: 10.1016/j.physrep.2009.11.002.

HJ, Lowe and G, Barnett. Understanding and using the medical subject headings (mesh) vocabulary to perform literature searches. *JAMA*, 271 (14):1103–1108, 1994. doi: 10.1001/jama.1994.03510380059038. URL +http://dx.doi.org/10.1001/jama.1994.03510380059038.

Hopcroft, John, Khan, Omar, Kulis, Brian, and Selman, Bart. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5249–5253, 2004. doi: 10.1073/pnas.0307750100. URL http://www.pnas.org/content/101/suppl.1/5249.abstract.

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: a review. *ACM Comput. Surv.*, 31(3): 264–323, September 1999. ISSN 0360-0300. doi: 10.1145/331499.331504. URL http://doi.acm.org/10.1145/331499.331504.

Jain, Anil K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010. ISSN 0167-8655. doi: 10.1016/j.patrec.2009.09.011. URL http://www.sciencedirect.com/science/article/pii/S0167865509002323. ¡ce:title¿Award winning papers from the 19th International Conference on Pattern Recognition (ICPR)¡/ce:title¿ ¡xocs:full-name¿19th International Conference in Pattern Recognition (ICPR)¡/xocs:full-name¿.

Leskovec, Jure, Kleinberg, Jon M., and Faloutsos, Christos. Graph evolution: Densification and shrinking diameters. *TKDD*, 1(1), 2007.

Newman, M. E. and Girvan, M. Finding and evaluating community structure in networks. *ArXiv*, 69(2): 026113, February 2004. doi: 10.1103/PhysRevE.69.026113.

Ng, Andrew Y., Jordan, Michael I., and Weiss, Yair. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 849–856. MIT Press, 2001.

Ruiz, Miguel E. and Srinivasan, Padmini. Hierarchical neural networks for text categorization (poster abstract). In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pp. 281–282, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1. doi: 10.1145/312624.312700. URL http://doi.acm.org/10.1145/312624.312700.

Schaeffer, Satu Elisa. Graph clustering. *Computer Science Review*, 1(1):27 – 64, 2007. ISSN 1574-0137. doi: 10.1016/j.cosrev.2007.05.001. URL http://www.sciencedirect.com/science/article/pii/S1574013707000020.

Zamir, Oren and Etzioni, Oren. Web document clustering: a feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pp. 46–54, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.290956. URL http://doi.acm.org/10.1145/290941.290956.

Zeng, Hua-Jun, He, Qi-Cai, Chen, Zheng, Ma, Wei-Ying, and Ma, Jinwen. Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pp. 210–217, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009030. URL http://doi.acm.org/10.1145/1008992.1009030.