A Large Benchmark Dataset for Web Document Clustering

Mark P. Sinka, David W. Corne
Department of Computer Science, University of Reading, Reading, RG6 6AY, UK
m.p.sinka@reading.ac.uk, d.w.corne@reading.ac.uk

Abstract: Targeting useful and relevant information on the WWW is a topical and highly complicated research area. A thriving research effort that feeds into this area is *document clustering*, which overlaps closely with areas usually known as *text classification* and *text categorisation*. A foundational aspect of such research (which has been proven over and over again in other research disciplines) is the use of standard datasets, against which different techniques can be properly benchmarked and assessed in comparison to each other. We note herein that, so far in this broad area of research, as many datasets have been used as research papers written, thus making it difficult to reason about the relative performance of different categorisation/clustering techniques used in different papers. In this paper we propose a standard dataset with a variety of properties suitable for a wide range of clustering and related experiments. We describe how the dataset was generated, and provide a pointer to it, and encourage its access and use. We also illustrate the use of part of the dataset by establishing benchmark results for simple *k*-means clustering, comparing the relative performance of k-means on a pair of 'close' categories and a pair of 'distant' categories. We naturally find that performance is better on the pair of 'distant' categories, however the experiments reveal that although stop-word removal is confirmed as helpful, word-stemming is, (perhaps counter to intuition), not necessarily always recommended on 'distant' categories.

1. Introduction

Most available search engines primarily function by providing a list of documents contained on the World Wide Web that contain matches to given keywords and/or phrases. However, keyword matching is known to be only suggestive of a document's relevance, and improvements to keyword matching need to be found in order for the World Wide Web to reach its full potential. To this end, considerable research is now being invested into more sophisticated ways to analyse and assess the content of web documents. For example, if the World Wide Web can be clustered into different subsets and labelled accordingly, search engine users can then restrict their keyword search to these specific subsets.

It seems fairly clear that we cannot expect manual classification to be achievable. The precise size of the World Wide Web is unknown and is growing all the time, however what is known is that Google [1] claims to index over 2 billion web documents, and according to [2], over 1.5 million web documents are added to the World Wide Web every day. Human back-classification of at least 2 billion web documents would be virtually impossible, and even attempting to categorize all new web documents would require unacceptable human effort. One possible solution is to

force web document authors to categorize each newly created page. This has three problems; firstly web authors cannot be relied upon to categorize their pages correctly, secondly authors are often prone to misclassify their documents in order to increase potential web traffic, and thirdly it does not address the problem of the (at least) 2 billion web pages that have already been created. Therefore researchers in this field are turning to autonomous, or semi-autonomous methods for web document categorization ([2], [3], [4], [5], [6], [7], [8], [9], [10]).

There are many other potential applications and benefits that will accrue from being able to reliably and *automatically* cluster and categorize corpora of documents. Much of this *document clustering* work is based on using either supervised or unsupervised learning techniques in order to label particular web documents as belonging to a specific category, or grouping together similar documents into clusters. This research area closely overlaps with (and in recent times indistinguishable from) research efforts known as *text classification* and *text categorization.* Van Rijsbergen [11] has carried out seminal work in this area, while excellent modern surveys have been done in [12] and [13], while [14] also provides a helpful tutorial.

Various techniques have been proposed that aim to develop accurate methods for autonomous categorization. However, the research literature in this area soon reveals that almost every single such proposed technique has been tested for its categorization accuracy using *different* datasets; any objective, scientifically sound comparison between two categorization techniques is therefore very difficult. This issue has also highlighted in [14]. Whilst it may be appealing to infer that a technique *X* that is shown to have a considerably superior accuracy than technique *Y* on dataset *A* would be similarly superior on dataset *B*, this is not really so clear cut. For example, stop-word removal [11] and stemming [15] are almost universally done to reduce the sizes of feature vectors for documents, in the belief that little information of use is lost (see, e.g., [6]); however, [16] shows counter-evidence to the general truth of this (as do we, later on). The essential point is that more widespread use of well-designed common datasets in this area will help elucidate all of the factors involved. For example, suppose dataset *A* is composed of two clusters which are relatively easy to separate, while dataset *B* may contain two very similar clusters, which are much harder to separate, and whatever strategy made technique *X* perform so well on dataset *A* may simply not be useful on dataset *B.* In the more general case, it is of course very hard to make general statements about two techniques that produce results that are *close*, in terms of accuracy, if they have not been tested on the same dataset.

The main aim of this paper is to propose a dataset for general use in web document clustering and similar experiments; the design, content, generation and location of this dataset are described in section 2. We note that this dataset cannot be specifically 'proven' to be optimally useful for web document clustering investigations, but it was designed with a view towards making it as flexible and useful as possible for such research and related work. In addition to the generation of a benchmark dataset, it is also important to establish benchmark values against which the accuracy of future techniques can be measured. Our particular research interest is the unsupervised clustering of web documents. Unsupervised clustering is attractive in this problem domain because, unlike supervised learning, it does not require a training set or domain expert in order to learn. This is important because firstly it is extremely difficult for any human or groups of humans to generate even a partial taxonomy for the World Wide Web, secondly any such taxonomy would be highly debatable and subject to question, and thirdly the fact that new categories emerge frequently and others diminish in importance would require such a taxonomy to be continually

updated. We are particularly interested in the ability of unsupervised methods to separate 'close' datasets, and in this paper we describe baseline experiments using straightforward *k*-means clustering, aimed at beginning to understand any interactions between the 'distance' of the categories being clustered, and aspects of the design of simple document feature vectors. Our *k*-means implementation, associated issues, and the experimental set-up are described in section 3, while the results of these experiments are listed in section 4. We briefly summarise and conclude in section 5.

2. A Benchmark Dataset

The datasets used by previous authors have varied a lot in terms of size and content. A dataset containing as few as 1,000 web documents was used [7], whereas in [8] 20,000 documents were used. The size of a dataset is important especially when considering unsupervised learning techniques such as *k*-means clustering, due to factors such as the increased sensitivity to initial cluster centres with a smaller dataset. Also there are many problems that are associated with web document feature extraction, such as the document representation, that are not as apparent in smaller datasets, which of course are less like the real world applications that we are attempting to address.

Human categorization of a dataset of sufficient size would take too long; hence researchers have employed previous human classification where it has been available. Humanly categorized sets of related web sites are called Web Directories, with the most common being Open Directory Project [17], Yahoo! Categories [18], and LookSmart [19]. Previous authors' datasets, despite taking advantage of these directories, have not extracted the data in the same way. The dataset generation method used in [7] involved picking 200 documents from 5 of the 14 top-level categories in Yahoo! [3]. This method of random web document selection not only meant that a high proportion of very small web documents were brought back, but also, as acknowledged in the paper, some documents belonged to multiple categories. This confounds human categorization, let alone automatic categorization.

Another important issue to consider is the number of categories contained within the dataset. A dataset comprising just 2 distinct categories would be much easier to partition into two clusters than would a dataset comprising 10 different categories, even if the latter dataset's categories fell into two clear themes. Our approach, as we see below, is to use a relatively large number of categories, but with sufficient of each category to allow a wide variety of sensible experiments, each using clearly defined subsets of the dataset.

Finally, as another argument towards the use of a standard dataset, we note that datasets used so far in document clustering research vary greatly in content. The same comparative experiments on two different datasets with different content, irrespective of size or the number of categories contained within, may easily produce conflicting results. The need for a common dataset that will allow for the accuracy of different techniques to be benchmarked is clear.

2.1 Dataset Design

The first distinct design decision was to ensure that each page in our dataset had a least a small amount of content. The task of autonomous categorization of medium to large

web documents is very complex and challenging, so it seems unwise to make this task even harder by attempting to classify documents containing very little or no information. Therefore our proposed dataset is biased towards pages that actually have some content, so a dataset that has as few 'very small' pages as possible should make developing and training of a classification/clustering system easier. It should be noted, therefore, that this dataset it not a realistic 'snapshot' of the World Wide Web.

The number of pages required for a useful, multi-purpose, dataset means that human classification of each web page in our dataset would be far too expensive in terms of time and effort. Therefore we made use of the Open Directory Project [17] and Yahoo! Categories [18] to provide web pages that have already been humanly categorized. Much thought then went into the selection of our dataset categories. The goal was to create a dataset consisting of some sets of categories quite distinct from each other, as well as other categories that were quite similar to each other. This allows for a range of clustering issues to be researched using the same dataset. The inclusion of similar categories also ensures that more complex partitioning tasks can be performed therefore fully testing the categorization accuracy of a particular method.

A suitable balance of categories was achieved by first selecting two broad but very distinct themes, namely "Banking & Finance" and "Programming Languages", and picking out three sub-categories from each of these two themes. The resulting six categories are: "Commercial Banks", "Building Societies", "Insurance Agencies", "Java", "C / C++" and "Visual Basic". It should be immediately clear that the task of partitioning these six categories into two groups, (Finance and Programming Languages) should be much easier than partitioning all the programming languages into three groups, which itself is easier than partitioning all of the six combined categories into six different groups. This therefore achieves our goal of generating a dataset allowing for various partitioning tasks with varying difficulty.

Four more categories were then selected, to widen the potential use and varieties of experiments that could be performed with this dataset. Since the themes "Banking & Finance" and "Programming Languages" are quite distinct, we decided it would be wise to include two more broad themes, one close to an existing theme, and one that was totally distinct from all categories so far. We therefore chose "Science" because it is *somewhat* close to "Programming Languages", and "Sport" because it is quite distinct from all the other categories in the database. The extra four categories which make up the main ten in our database were therefore: "Astronomy", "Biology", "Soccer" and "Motor Sport".

Finally, we felt I useful to add an extra category that was in some way a 'parent' of two existing ones. This provides a good test for hierarchical clustering methods, and for generally reasoning about the results of clustering in terms of hierarchical relationships among the data. The extra dataset is "Sport", and contains web documents from all the sites that were classified as sport (in [17] and [18]), but with the sites that occurred in either the "Soccer" or "Motor Sport" datasets removed. This allows us to think of quite complex partitioning tasks, such as the task of partitioning the union of sets "Sport", "Soccer" and "Astronomy" into two groups. A full list of the 11 main dataset categories is shown in Table 1.

In order to determine the number of documents required to make our dataset useful we looked at not only the size of dataset used by previous researchers, but the number of documents per category also. Knowing that our dataset consisted of 11 categories, the decision was whether to download and archive 100, 200, 500, 1,000 or 2,000 documents per category. We felt that 1,000 documents per category would

suffice, therefore giving an overall dataset size of 11,000 documents. The next subsection details how we went about extracting the pages themselves.

Table 1 **– Dataset categories and their associated themes.**

Dataset Id	Dataset Category	Associated Theme
A	Commercial Banks	Banking & Finance
B	Building Societies	Banking & Finance
C	Insurance Agencies	Banking & Finance
D	Java	Programming Languages
E	C / C++	Programming Languages
F	Visual Basic	Programming Languages
G	Astronomy	Science
H	Biology	Science
I	Soccer	Sport
J	Motor Sport	Sport
K	Sport	Sport

2.2 Page Selection

For each of the first 10 main categories in table 1, we extracted the set of all websites contained within the associated category listing in the Open Directory Project [17], and combined them with the set of all websites contained within the associated category listing on Yahoo! [18]. The only information stored about each web site was the entry page, which allowed our web spider to crawl the rest of the site. Each site in the database was then crawled, noting down the size and URL of each page. The size of a page was determined by first removing any scripts, style-sheets, or comments beforehand. Each site's URLs were then ordered, in descending size of the referenced page, and stored in the database. The next step was to remove any sites that contained fewer than 10 pages in total. This was done because of the way in which the URLs are archived.

Table 2 – Attributes saved within each web document

Id number
Archiving date and time
Page size
Corresponding Category (Content label)
HTML Source

The archiving involved looping through each web site in the database and 'popping' off the largest web page's URL from that site. The contents of the URL were then retrieved and examined for the presence of a frameset. If the page was, or contained a frameset then the rest of the frames pointed to were added to the current page before archiving. This ensured the actual content (as seen by a human) was archived. Each complete page was then archived with the content indicated in table 2.

We freely encourage the use of the dataset and a complete version can be downloaded from: http://www.pedal.reading.ac.uk/banksearchdataset/

3. Baseline Experiments with *K*-Means Clustering

Some experiments were done which are now described; these experiments had two aims. First, to establish some baseline results against which future techniques applied to the same dataset can be evaluated. Second, these experiments form the beginning of our study into appropriate techniques for the unsupervised clustering of documents that are contained in 'close' categories. This is a particularly interesting problem which relates to, among other things, technologies which can automatically discern fine-grained differences within document sets and (for example) add extra value to the output of search engines.

We use only the most basic, although popular and effective in many areas, unsupervised clustering technique, *k*-means, and also use a basic approach to developing feature vectors to represent the documents (simple word-frequency vectors). These choices served both of the aims as follows. First, more sophisticated techniques can only be properly evaluated if results are available for simpler methods on the same data – indeed, simple *k*-means may well offer the best speed/performance trade-off on certain datasets. Second, although a number of sophisticated and interesting techniques are currently under study in unsupervised document clustering ([20], [21], [22], [10]), this field is far less researched so far than *supervised* document categorisation. This heightens the importance, in our view, of thorough study of rudimentary techniques in the field, especially given the lack of dataset standardisation so far, and the fact that little or nothing seems to have been done concerning the unsupervised separation of close categories.

Towards understanding *k*-means performance on 'close' category discrimination, we did two sets of experiments, both with *k* fixed appropriately at 2. One set used *k*–means to separate two semantically very similar categories of documents: sets *B* and *C* (see Table 1). The other set of experiments used *k*-means to separate two quite distinct categories: sets *A* and *I* (see Table 1). In all cases, the full 1,000 documents in the dataset for each of the two categories were used. Hence, each experiment attempted to cluster 2,000 documents. In each of the two main sets of experiments, we did 10 trial runs for each of 16 different feature vector configurations as follows. The feature vector representing a document (as described next in more detail) is a simple vector of scaled word frequencies. However, the vector was built either with or without stemming [15], and either with or without removal of stop-words [11]. Also, the vector was built only using the top *h*% of words (in the 2,000 documents being clustered) in terms of frequency of occurrence. We tested four values of *h*: 0.5, 1, 1.5, and 2. Results are presented in section 4, but before that we give more detail of the experimental set-up and our *k*-means implementation.

3.1 Feature Extraction

Stressing simplicity first, our feature vectors were built only from text that would be seen on the screen, i.e. normal document text, image captions and link text, and no

extra weight was given according to emphasis (bold typeface, italic typeface, different colours, etc …). For each document, the extraction process was as follows:

- The set of all words that appeared at least once in the document was extracted.
- If stop-word removal was switched on, we removed from the set of extracted words (Step 1) any word that was listed in our stop-word list, (we used Van Rijsbergen's list [11]).
- If word stemming was switched on, we combined all the words with a similar stem, (i.e. count all occurrences of a word as a single occurrence of its stem).
- We then recorded and stored the frequency of each word in that document.

Once all the documents in the chosen set of categories were thus processed, the next step was to create a master word list that containing every word in the combined dataset, associated with its overall frequency. Then we cut down the master list to contain only the top h% of most frequently occurring words, where h was varied between experiments. Finally, a feature vector $\mathbf{v}_i$ was created for each document i, such that the j^{th} element in $\mathbf{v}_i$ was w_{ji}/s_i, where w_{ji} is the number of occurrences in document i of the j^{th} most frequent word in the combined dataset, and s_i is the total number of words in document i.

3.2 Clustering

Our implementation of k-means clustering was standard, although certain issues tend to vary between implementations and we clarify those here. K-means begins by generating random vectors to act as initial cluster centres. Each required cluster centre was created by coping the contents of a randomly chosen document feature vector from the vector space. Another important issue is the treatment of 'dead' cluster centres (containing no documents, since all vectors are closer to some other cluster centre). We chose to do nothing when a cluster centre died, in case a vector became re-assigned to it in the future. Again, this stressed simplicity, although other options are generally more favourable, especially when k is low.

4. Results

The results are summarised in Table 3. For each configuration, three measurements are given – these are the mean, median, and best accuracies over ten trials for that configuration. We measure the accuracy of a single clustering run as the percentage of documents that were classified correctly. That is, if 2-means clustering happens to separate the *A* & *I* dataset into two clusters of 1,000 documents, with all the *A* in one cluster and all the *I* in the other, that represents 100% accuracy. More generally, a result would place X documents from a category into one cluster, and the remaining Y of those documents into the other cluster.

Table 3 **–** Experiment Results: mean, median and best of ten runs for two sets of sixteen experiments each using different configurations of frequency vector size, stop-word removal, and stemming. A and I are the dissimilar datasets and B and C are similar datasets (see Table 1). The highest mean, median and best for each experiment is underlined, and the best accuracy for each set (A & I, B & C) is in bold type.

Datasets	Stemming / Stop-word removal	Accuracy	Size of word-frequency vector			
			0.5%	1%	1.5%	2%
A & I	No/No	Mean	57.12	56.31	59.64	53.2
		Median	56	50.3	64.9	50.77
		Best	64.95	66.4	66.3	66.3
A & I	No/Yes	Mean	62.38	66.65	71.12	58.8
		Median	51.07	50.875	71.7	50.025
		Best	90.65	92.55	93.0	**93.05**
A & I	Yes/No	Mean	59.8	58.27	60.51	58.75
		Median	59.2	50.8	60.0	50.8
		Best	69.4	71.1	71.15	71.6
A & I	Yes/Yes	Mean	53.93	65.31	54.25	67.51
		Median	50.15	50.75	50.05	55.6
		Best	87.1	87.8	91.95	91.35
B & C	No/No	Mean	53.66	53.79	50.68	50.6
		Median	51.9	51.8	50.65	50.05
		Best	59.9	58.2	51.7	51.7
B & C	No/Yes	Mean	53.67	53.52	51.44	52.36
		Median	54.3	54.3	51.6	53.03
		Best	54.7	54.3	52.75	54.4
B & C	Yes/No	Mean	53.7	51.01	50.69	51.55
		Median	52.1	51.0	50.15	51.95
		Best	62.15	52.1	51.95	51.95
B & C	Yes/Yes	Mean	70.9	75.44	64.19	69.22
		Median	74.02	89.0	56	60.9
		Best	89.1	90.05	89.2	**90.35**

We then took the cluster containing the larger number of documents to be the correct cluster, and calculated accuracy on that basis. Many experiments yielded poor results in which all or nearly all documents ended up in a single cluster, which nevertheless scores 50% on this accuracy measure.

Throughout the experimental trials, there were many poor results, owing to the sensitivity of (straightforward) k-means to the positioning of the initial random cluster centres. This is apparent in the low mean and median values; such sensitivity was particularly problematic in those experiments which show low medians (near 50%), which indicate that more than half of the ten trials places all documents within a single cluster. However, sufficient trials were done to show interesting effects as follows.

Predictably, we find that *k*-means can more readily separate the distinct categories (*A* and *I*), with good results appearing when stop-word removal was in force. Unexpectedly, however, in the distinct-categories case, although stemming was more effective than its absence, stemming in conjunction with stop-word removal gave slightly worse results than stop-word removal alone. With highly similar categories, results were generally poor except when both stop-word removal and stemming were used. Both were necessary to achieve any reasonable degree of separation – although, when separation was achieved, the best results were around 90% accuracy, which compares well with the better results from the distinct-category case.

Finally, there are no clear trends concerning the size of the feature vector. There is a bias towards longer feature vectors giving slightly better results, but accurate and reliable results often occurred at the smallest size. Naturally, the larger the feature vector the more words taken into account, but each increase in the feature vector size 'captures' some words useful for discrimination and others not useful (and perhaps positively unhelpful). It is nevertheless interesting that fairly small feature vectors can produce accurate results, even using straightforward *k*-means and word-frequency representation, on the similar-category clustering task. In both cases, the full word list was around 4,000 words, so the smaller feature vectors were of length *c*. 20, and the 1% experiments (length *c*. 40) were able to separate similar categories fairly well.

5. Conclusion

Web Document clustering is an exciting and thriving research area. In particular, unsupervised clustering of web documents – so far less studied than supervised learning in this context – has many future applications in organising and understanding the WWW, as well as other corpora of text. We have generated a dataset to support our own ongoing work in this area, and encourage its general use by other researchers. The dataset is especially designed to support a wide range of experiments in unsupervised clustering, but naturally supports supervised learning too.

We have shown some benchmark results for k–means applied to subsets of our dataset, with different configurations of feature vector, and with a view to investigating the use of unsupervised clustering to separate datasets which are only finely distinct. Surprisingly, we found that stemming seems not to be universally helpful (although it is generally helpful), with stop-word-removal alone sufficient for good separation by *k*-means of distinct categories using a small word-frequency feature vector; stemming in *addition* slightly worsened results in this case. An intriguing overall finding was that a pair of very similar categories could also be separated quite well by straightforward *k*-means, but in this case reasonable results only appeared when both stemming and stop-word removal were applied.

6. Acknowledgements

The authors wish to thank SEEDA (the South East England Development Agency), the EPSRC (Engineering and Physical Sciences Research Council), and Simon Anderson and others at BankSearch Information Consultancy Ltd, for their ongoing financial support for this research.

References

[1] Google Search Engine, <http://www.google.com>

[2] Pierre, J.M. (2000) *Practical Issues for Automated Categorization of Web Sites*, September 2000. <http://citeseer.nj.nec.com/pierre00practical.html>

[3] Boley, D., Gini, M., Gross, R., Han, S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moor, J. (1999a) Document Categorization and Query Generation on the World Wide Web Using WebACE, *AI Review*, **13**(5–6): 365–391,

[4] Boley, D., Gini, M., Gross, R., Han, S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moor, J. (1999b) Partitioning-Based Clustering for Web Document Categorization, *Decision Support Systems*, **27**(3): 329–341, <http://citeseer.nj.nec.com/9105.html>

[5] Goldszmidt, M., Sahami, M. (1998) *A Probabilistic Approach to Full-Text Document Clustering*, Technical Report ITAD-433-MS-98-044, SRI International, <http://citeseer.nj.nec.com/goldszmidt98probabilistic.html>

[6] Moore, J., E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. (1997) Web page categorization and feature selection using association rule and principal component clustering. In *7th Workshop on Information Technologies and Syst*ems, Dec 1997. <http://citeseer.nj.nec.com/moore97web.html>

[7] Tsukada, M., Washio, T. and Motoda, H. (2001) Automatic Web-Page Classification by Using Machine Learning Methods, in N. Zhong, Y. Yao, J. Liu, S. Oshuga (eds.) *Web Intelligence: Research and Development, Proceedings of the 1st Asia Pacific Web Conference on Web Intelligence*, LNAI 2198, Springer-Verlag, Berlin, pp. 303–313

[8] Wong, W., Fu, A.W. (2000) Incremental Document Clustering for Web Page Classification, *IEEE 2000 Int. Conf. on Info. Society in 21st century: emerging technologies and new challenges*, Nov 5-8, 2000, Japan.<http://citeseer.nj.nec.com/article/wong01incremental.html>

[9] Zamir, O., Etzioni, O. (1998) Web document clustering: A feasibility demonstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 46--54, Melbourne, Australia.

[10] Zhao, Y., Karypis, G. (2002) C*riterion Functions for Document Clustering: Experiments and Analysis*, <http://citeseer.nj.nec.com/zhao02criterion.html>

[11] Van Rijsbergen, C.J. (1975) *Information Retrieval*, Butterworths.

[12] Aas, K., Eikvil, A. (1999) *Text Categorisation: A survey*, Technical report, Norwegian Computing Center, June, <http://citeseer.nj.nec.com/aas99text.html>

[13] Sebastiani, F. (1999a) *Machine learning in automated text categorisation: A survey*. Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione, C.N.R., Pisa, IT, 1999. . <http://citeseer.nj.nec.com/article/sebastiani99machine.html>

[14] Sebastiani, F. (1999b) A Tutorial on Automated Text Categorisation. In Analia Amandi and Ricardo Zunino, editors, *Proceedings of ASAI-99, 1st Argentinean Symposium on Artificial Intelligence*, pages 7–35, Buenos Aires, AR, 1999. <http://citeseer.nj.nec.com/sebastiani99tutorial.html>

[15] Porter, M.F. (1980), An algorithm for suffix stripping, *Program*, **14**(3): 130–137.

[16] Riloff, E. (1997) Little words can make a big difference for text classification. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 130–136. <http://citeseer.nj.nec.com/riloff95little.html>

[17] Open Directory Project, <http://www.dmoz.org>

[18] Yahoo! Directory, <http://www.yahoo.com>

[19] LookSmart, <http://www.looksmart.com>

[20] Tishby, N., Pereira, F.C., Bialek, W. (1999) The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 368-377, 1999. <http://citeseer.nj.nec.com/tishby99information.html>

[21] Slonim, N. and Tishby, N. (2000) Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 208–215.

[22] Slonim, N., Tishby, N. (2001) The power of word clusters for text classification. In *23rd European Colloquium on Information Retrieval Research*. <http://citeseer.nj.nec.com/slonim01power.html>