# Filtering Speaker-Specific Words from Electronic Discussions

**Ingrid Zukerman** and **Yuval Marom**
School of Computer Science and Software Engineering
Monash University, Clayton, Victoria 3800, AUSTRALIA
{ingrid, yuvalm}@csse.monash.edu.au

## Abstract

The work presented in this paper is the first step in a project which aims to cluster and summarise electronic discussions in the context of help-desk applications. The eventual objective of this project is to use these summaries to assist help-desk users and operators. In this paper, we identify features of electronic discussions that influence the clustering process, and offer a filtering mechanism that removes undesirable influences. We tested the clustering and filtering processes on electronic newsgroup discussions, and evaluated their performance by means of two experiments: *coarse-level clustering* and *simple information retrieval*. Our evaluation shows that our filtering mechanism has a significant positive effect on both tasks.

## 1 Introduction

The ability to draw on past experience is often useful in information-providing applications. For instance, users who interact with help-desk applications would benefit from the availability of relevant contextual information about their request, e.g., from previous, similar interactions between the system and other users, or from interactions between domain experts.

The work reported in this paper is the first step in a project which aims to provide such information. The eventual objective of our project is to automatically identify related interactions in help-desk applications, and to generate summaries from their combined experience. These summaries would then assist both users and operators.

Our approach to the identification of related interactions hinges on the application of clustering techniques. These techniques have been used in Information Retrieval for some time (*e.g.* Salton, 1971). They involve grouping a set of related documents, and then using a representative element to match input queries (as opposed to matching the whole collection of documents). Document clustering has been used in search engine applications to improve and speed up retrieval (*e.g.* Zamir and Etzioni, 1998), but also for more descriptive purposes, such as using representative elements of a cluster to generate lists of keywords (Neto et al., 2000).

However, discussions (and dialogues in general) have distinguishing features which make clustering a corpus of such interactions a more challenging task than clustering plain documents. These features are: (1) the corpus consists of contributions made by a *community* of authors, or "speakers"; (2) certain speakers are more dominant in the corpus; and (3) speakers often use idiosyncratic, speaker-specific language, or make comments that are not about the task at hand.

In this paper, we report on a preliminary study where we cluster discussions carried out in electronic newsgroups. Specifically, we report on the influence of the above features on the clustering process, and describe a filtering mechanism that identifies and removes undesirable influences.

Table 1 shows the newsgroups used as data sources in our experiments. These newsgroups were obtained from the Internet. The table shows the number of threads in each newsgroup, the number of people posting to the newsgroups, and the highest number of postings by an individual for each newsgroup. It also shows the impact of the filtering mechanism on each newsgroup (Section 2).

The clustering process and filtering mechanism were evaluated by means of two experiments: (1) coarse-level clustering, and (2) simple information retrieval.

**Coarse-level clustering.** This experiment consists of merging the discussion threads (documents) in different newsgroups into a single dataset, and applying a clustering mechanism to separate them. The performance of the clustering mechanism is then evaluated by how well the generated clusters match the original newsgroups from which the discussion threads were obtained. Clearly, this evaluation is at a coarser level of granularity than that required for our final system. However, we find it useful for the following reasons:

| newsgroup | number of threads | number of people | most frequent number of postings | average filter usage (per thread) |
|---|---|---|---|---|
| lp.hp | 1920 | 1707 | 715 (17.5%) | 0.5 |
| comp.text.tex | 1383 | 1140 | 246 (4.6%) | 7.4 |
| comp.graphics.apps.photoshop | 1637 | 1586 | 395 (5.8%) | 6.9 |

Table 1: Description of the newsgroups.

- Owing to the number and diversity of newsgroups on the Internet, we can perform controlled experiments where we vary the degree of similarity between newsgroups, thereby simulating discussions with different levels of relatedness.
- Our experiments show that our filtering mechanism has a positive influence at different levels of granularity (Section 4). Hence, there is reason to expect that this influence will remain for finer levels of granularity, e.g., the level of a task or request.
- Finally, the different newsgroups are identified in advance, which obviates the need for manual discussion-tagging at this stage.

Due to space limitations, we report only on a subset of our experiments. In (Marom and Zukerman, 2004) we present a comparative study that considers different sets of newsgroups of varying levels of relatedness. We regard the set of newsgroups presented here as having a "medium" level of relatedness.

**Simple information retrieval.** This experiment constitutes a simplistic and restricted version of the document retrieval functionality envisaged for our eventual system. In this experiment, we matched pairs of query terms to the centroids of the generated clusters, and assessed the system's ability to retrieve relevant discussion threads from the best-matching cluster, with and without filtering. The experiment makes the implicit assumption that the corpus contains discussions relevant to incoming requests, *i.e.* that new requests are similar to old ones. We believe that the results of this restricted experiment are indicative of future system performance, as the envisaged system is also expected to operate under this assumption.

Next, we describe our filtering mechanism. Section 3 describes the clustering procedure, including our data representation and cluster identification method. Section 4 presents the results from our experiments, and Section 5 concludes the paper.

## 2 The Filtering Mechanism

Our filtering mechanism identifies and removes idiosyncratic words used by dominant speakers. Such words typically have a high frequency in the postings authored by these speakers. Even though these words can appear anywhere in a person's posting, they appear mostly in signatures (about 75% of these words appear towards the end of a person's posting, while the remaining 25% are distributed throughout the posting). We therefore refer to them throughout this paper as *signature words*.

The filtering mechanism operates in two stages: (1) profile building, and (2) signature-word removal.

**Profile building.** First, our system builds a 'profile', or distribution of *word posting frequencies*, for each person posting to a newsgroup. The posting frequency of a word is the number of postings where the word is used. For example, a person might have two postings in one newsgroup discussion, and three postings in another, in which case the maximum possible posting frequency for each word used by this person is five. Alternatively, one could count all occurrences of a word in a posting, which could be useful for constructing more detailed stylistic profiles. However, at present we are mainly concerned with words that appear across postings.

**Signature-word removal.** In the second stage, word-usage proportions are calculated for each person. These are the word posting frequencies divided by the person's total number of postings. The aim of this calculation is to filter out words that have a very high proportion. In addition, we wanted to distinguish between the profile of a dominant individual and that of a non-dominant one. Hence, rather than just using a simple cut-off threshold for word-usage proportions, we base the decision to filter on the number of postings made by an individual as well as on the proportions. This is done by utilising a statistical significance test (a Bernoulli test) that measures if a proportion is *significantly* higher than a threshold (0.4),[1] where significance is based on the number of postings.

The impact of this filtering mechanism on the various newsgroups is shown in the last column of Table 1, which displays the average number of times the filter is applied per discussion thread. This number gives an indication of the existence of signature

[1] Although this threshold seems to pick out the signature words, we have found that the filtering mechanism is not very sensitive to this parameter. That is, its actual value is not important so long as it is sufficiently high.

words from dominant speakers. For example, although the 'hp' newsgroup has a very dominant individual (who accounts for 17.5% of the postings), the filter is applied to this person's postings a very small number of times, as s/he does not have signature words. In contrast, the 'tex' and 'photoshop' newsgroups have less dominant individuals, but here the filter is applied more frequently, as these individuals do have signatures.

## 3 The Clustering Procedure

The clustering algorithm we have chosen is the K-Means algorithm, because it is one of the simplest, fastest, and most popular clustering algorithms. Further, at this stage our focus is on investigating the effect of the filtering mechanism, rather than on finding the best clustering algorithm for the task at hand. K-Means places $k$ centers, or *centroids*, in the input space, and assigns each data point to one of these centers, such that the total Euclidean distance between the points and the centers is minimised.

Recall from Section 1 that our evaluative approach consists of merging discussion threads from multiple newsgroups into a single dataset, applying the clustering algorithm to this dataset, and then evaluating the resulting clusters using the known newsgroup memberships. Before describing how clusters created by K-Means are matched to newsgroups (Section 3.2), we describe the data representation used to form the input to K-Means.

### 3.1 Data Representation

As indicated in Section 1, we are interested in clustering complete newsgroup *discussions* rather than individual postings. Hence, we extract discussion threads from the newsgroups as units of representation. Each thread constitutes a document, which consists of a person's inquiry to a newsgroup and all the responses to the inquiry.

Our data representation is a bag-of-words with TF.IDF scoring (Salton and McGill, 1983). Each document (thread) yields one data point, which is represented by a vector. The components of the vector correspond to the words chosen to represent a newsgroup. The values of these components are the normalised TF.IDF scores of these words.

The words chosen to represent a newsgroup are all the words that appear in the newsgroup, except function words, very frequent words (whose frequency is greater than the 95th percentile of the newsgroup's word frequencies), and very infrequent words (which appeared less than 20 times throughout the newsgroup). This yields vectors whose typical dimensionality (*i.e.* the number of words retained) is between 1000 and 2000. Since dimensionality reduction is not detrimental to retrieval performance (Schütze and Pedersen, 1995) and speeds up the clustering process, we use Principal Components Analysis (Affi and Clark, 1996) to reduce the dimensionality of our dataset. This process yields vectors of size 200.

The TF.IDF method is used to calculate the score of each word. This method rewards words that appear frequently in a document (term frequency – TF), and penalises words that appear in many documents (inverse document frequency – IDF). There are several ways to calculate TF.IDF (Salton and McGill, 1983). In our experiments it is calculated as $\text{TF}_{ij} = \log_2(f_{ij} + 1)$ and $\text{IDF}_i = \log_2(N/n_i)$, where $f_{ij}$ is the frequency of word $i$ in document $j$, $n_i$ is the number of documents where word $i$ appears, and $N$ is the total number of documents in the dataset. In order to reduce the effect of document length, the TF.IDF score of a word in a document is then normalised by taking into account the scores of the other words in the document.

One might expect that the IDF component should be able to reduce the influence of signature words of dominant individuals in a newsgroup. However, IDF alone cannot distinguish between words that are representative of a newsgroup and signature words of frequent contributors, *i.e.* it would discount these equally. Further, we have observed that an individual does not have to post to many threads (documents) for his/her signature words to influence the clustering process. Since IDF discounts words that occur in many documents, it would fail to discount signature words that appear mainly in the *subset* of documents where such individuals have postings.

### 3.2 Clustering and Identification

In order to evaluate the clusters produced by K-Means for a particular dataset, we compare each document's cluster assignment to its true 'label' — a value that identifies the newsgroup to which the document belongs, of which there are $l$ (three in the dataset considered here). However, because K-Means is an unsupervised mechanism, we do not know which cluster to compare with which newsgroup. We resolve this issue as follows.

We calculate the goodness of the match between each cluster $i \in \{1..k\}$ and each newsgroup $j \in \{1..l\}$ ($k \geq l$) using the F-score from Information Retrieval (Salton and McGill, 1983). This gives an overall measure of how well the cluster represents the newsgroup, taking into account the 'correctness' of the cluster (precision) and how much of the newsgroup it accounts for (recall). Precision is calculated

as

$$P_{ij} = \frac{\text{\# documents in cluster } i \text{ and newsgroup } j}{\text{\# documents in cluster } i}$$

and recall as

$$R_{ij} = \frac{\text{\# documents in cluster } i \text{ and newsgroup } j}{\text{\# documents in newsgroup } j}$$

The F-score is then calculated as

$$F_{ij} = \{0.5(\frac{1}{P_{ij}} + \frac{1}{R_{ij}})\}^{-1}$$

Once all the $F_{ij}$ have been calculated, we choose for each cluster the best newsgroup assignment, *i.e.* the one with the highest F-score. As a result of this process, multiple clusters may be assigned to the same newsgroup, in which case they are pooled into a single cluster. The F-score is then re-calculated for each pooled cluster to give an overall performance measure for these clusters.

The clustering procedure is evaluated using two main measures: (1) the number of newsgroups that were matched by the generated clusters (between 1 and $l$), and (2) the F-score of the pooled clusters. The first measure estimates how many clusters are needed to find all the newsgroups, while the second measure assesses the quality of these clusters. Further, the number of clusters that are needed to achieve an acceptable quality of performance suggests the level of granularity needed to separate the newsgroups (few clusters correspond to a coarse level of granularity, many clusters to a fine one).

The clustering procedure is also evaluated as a whole by calculating its overall precision, *i.e.* the proportion of documents that were assigned correctly over the whole dataset. Note that the overall recall is the same as the overall precision, since the denominators in both measures consist of all the documents in the dataset. Hence, the F-score is equal to the precision.

### 3.3 Example

We now show a sample output of the clustering procedure described above, with and without the filtering mechanism described in Section 2. Tables 2 and 3 display the pooled clusters created without and with filtering, respectively. These tables show how many clusters were found for each newsgroup, the number of documents in each pooled cluster, and the performance of the cluster (P, R and F). The tables also present the top 30 representative words in each cluster (restricted to 30 due to space limitations). These words are sorted in decreasing order of their average TF.IDF score over the documents in the cluster (words representative of a cluster should have high TF.IDF scores, because they appear frequently in the documents in the cluster, and infrequently in the documents in other clusters).

According to the results in Table 2, the top-30 list for the 'hp' cluster does not have many signature words. This was anticipated by the observation that the filtering mechanism was applied very rarely to the 'hp' newsgroup (Table 1). In contrast, the majority of the top-30 words in the 'tex' cluster are signature words (some exceptions are 'chapter', 'english' and 'examples'). We conclude that this pooled cluster was created (using two different clusters) to represent the various signatures in the 'tex' newsgroup. Further, a relatively small number of documents are assigned to the 'tex' cluster, which therefore has a very low recall value (0.34). Its precision is perfect, but its low recall suggests that many of the documents representing the true topics of this newsgroup were assigned to other clusters.

The 'photoshop' cluster has a very high precision and recall, so most of the 'photoshop' documents were assigned correctly. However, here too many of the top words are signature words. Even when the 'obvious' signature words are ignored (such as URLs and people's names), there are still words that confuse the topics of this newsgroup, such as 'million', 'america', 'urban' and 'dragon'.

In Table 3 most of the words discovered by the clustering procedure represent the true topics of the newsgroups. The filtering mechanism removes the dominant signature words, and thus the clustering procedure is able to find the true topic-related clusters (precision and recall are very high for all pooled clusters). Notice that there are still some signature-related words, such as 'arseneau' and 'fairbairns' in the 'tex' cluster, and 'tacit' and 'gifford' in the 'photoshop' cluster. These words correspond mainly to a dominant individual's name or email address, and the filtering mechanism fails to filter them when *other* individuals reply to the dominant individual using these words. In a thread (document) containing a dominant individual, that individual's signature words are filtered, but unless the people replying to the dominant individual are dominant themselves, the words they use to refer to this individual will *not* be filtered, and therefore will influence the clustering process. This highlights further the problem that our filtering mechanism is addressing, and suggests that more filtering should be done.

## 4 Evaluation

The example presented in the last section pertains to a specific run of the clustering procedure. We now evaluate our system more generally by looking at

**hp** (1 cluster, 1825 documents, P=0.58, R=0.97, F=0.73)
unable, connected, hat, entry, fix, configure, lpd, configuration, parallel, psc, kernel, configured, kurt, de, taylor, report, local, asnd@triumf.ca, grant, plain, debian, linuxprinting.org, officejet, instructions, letter, appears, update, called, extra, compile

**tex** (2 clusters, 375 documents, P=1.00, R=0.34, F=0.50)
luecking, arkansas, http://www.tex.ac.uk..., herbert, piet, oostrum, university, heiko, lars, mathematical, department, voss, van, http://people.ee.eth..., sciences, madsen, rtfsignature, http://www.ctan.org/..., http://www.ams.org/t..., wilson, oberdiek, http://www.ctan.org/..., apr, examples, english, asnd@triumf.ca, chapter, rf@cl.cam.ac.uk, sincerely, private

**photoshop** (2 clusters, 1143 documents, P=0.95, R=0.95, F=0.95)
gifford, million, jgifford@surewest.ne..., heinlein, www.nitrosyncretic.c..., john@stafford.net, america, urban, dragon, fey, imperial, created, hard, pictures, rgb, edjh, folder, face=3darial, tutorials, professional, comic, graphic, sketches, http://www.sover.net..., move, drive, wdflannery@aol.com, colors, buy, posted

Table 2: Top 30 centroid words found by the clustering procedure *without* filtering.

**hp** (2 clusters, 1162 documents, P=0.97, R=0.93, F=0.95)
lprng, connected, linuxprinting.org, kernel, red, psc, hat, configure, unable, configuration, configured, parallel, ljet, printtool, series, database, jobs, gimp-print, debian, entry, suse, cupsomatic, officejet, cat, perfectly, jetdirect, duplex, devices, kde, happens

**tex** (1 cluster, 1040 documents, P=0.98, R=0.91, F=0.95)
arseneau, ctan, fairbairns, style, miktex, pdflatex, faq, chapter, apr, symbols, dvips, figures, title, include, math, bibtex, kastrup, university, examples, english, dvi, peter, plain, documents, contents, written, e.g, macro, robin, donald

**photoshop** (2 clusters, 1287 documents, P=0.88, R=0.98, F=0.93)
tacit, james, gifford, folder, rgb, pictures, created, colors, tutorials, illustrator, window, tom, mask, money, whatever, newsgroup, drive, brush, plugin, professional, stafford, view, menu, palette, channel, graphic, pixel, ram, tutorial, paint

Table 3: Top 30 centroid words found by the clustering procedure *with* filtering.

clustering performance for a range of values of $k$, and inspecting the implications of this performance with respect to a document retrieval task.

### 4.1 Coarse-Level Clustering

Figure 1 shows the overall clustering performance obtained without filtering (solid line) and with filtering (dashed line). The left-hand-side of the figure shows the average number of newsgroups matched to clusters, while the right-hand-side shows the overall performance (F-score) obtained. The error bars in the plots are averages of 100 repetitions of the clustering procedure described in Section 3.2 (with random initialisation of the centroids at the start of each run). The widths of the error bars indicate 95% confidence intervals for these averages. Hence, non-overlapping intervals correspond to a difference with p-value lower than 0.05.

In (Marom and Zukerman, 2004), we show that the effect of the filtering mechanism on clustering performance depends on three factors: (1) the presence of signature words from dominant contributors; (2) the 'natural', topical overlap between the newsgroups; and (3) the level of granularity in the clustering, *i.e.* the number of centroids.

The main conclusions with respect to the dataset presented here are as follows.

- Firstly, there is a heavy presence of signature words in two of the newsgroups ('tex' and 'photoshop' – see Table 1), and therefore the filtering mechanism has a significant effect on this dataset as a whole. As can be seen in Figure 1, the performance (F-score) without filtering is poorer for all values of $k$, and substantially more so for low values of $k$. Although the clustering procedure without filtering is able to find three distinct newsgroups with $k = 5$, it requires a higher value of $k$ to achieve a satisfactory performance. This suggests that the signature words create undesirable overlaps between the clusters. In contrast, when filtering is used, the clustering procedure reaches its best performance with $k = 4$, where the performance is extremely good.
- Secondly, the fact that the performance with filtering converges for such a low value of $k$ suggests that there is little true topical overlap between the newsgroups, and the fact that the performance is significantly better for $k = 4$

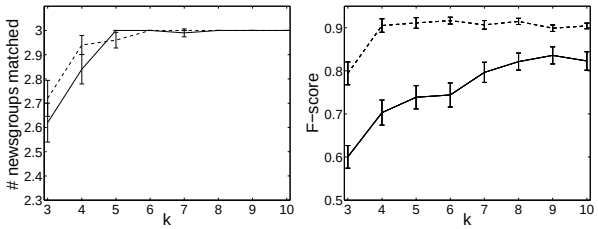


Figure 1: Overall clustering performance.

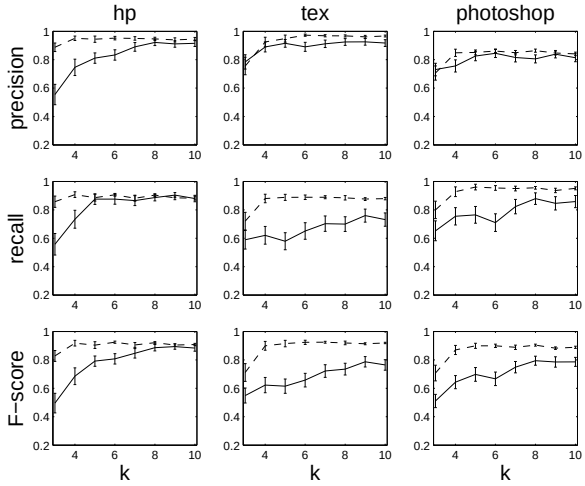


Figure 2: Clustering performance by newsgroup.

than for $k = 3$ suggests that there is some overlap, possibly created by a sub-topic of one of the newsgroups. That is, although there are only three newsgroups, four centroids are better at finding them than three centroids, because the fourth centroid may correspond to an overlap region between two clusters, which then gets assigned to the correct newsgroup.

We can get a better insight into these results by inspecting the individual performance of the pooled clusters, particularly their precision and recall. Figure 2 shows the average performance of the pooled clusters separately for each of the three newsgroups. This figure confirms that the 'hp' newsgroup is the least affected by signature words: for low values of $k$, without filtering, the average performance (F-score) of the pooled clusters corresponding to the 'hp' newsgroup is generally better than that of the clusters corresponding to the other newsgroups (and it even matches the performance achieved *with* filtering for $k = 9$). This is particularly evident when we compare recall curves: recall for the 'hp' newsgroup without filtering reaches the recall obtained with filtering when $k = 5$. In contrast, precision only achieves this level of performance for higher values of $k$ — this is because some of the documents in the 'hp' newsgroup are confused with documents in the other two newsgroups.

### 4.2 Simple Information Retrieval

A desirable outcome for retrieval systems that perform document clustering prior to retrieval is that the returned clusters contain as much useful information as possible regarding a user's query. If the clustering is performed well, the words in the query should appear in many documents in the best matching cluster(s).

Our retrieval experiments consist of retrieving documents that match three simple queries, each comprising a word pair that occurs frequently in the newsgroups. As before, for each experiment we repeated the clustering procedure 100 times and averaged the results. Retrieval performance was measured as follows:

$$\frac{\text{correct documents in the selected cluster}}{\text{total correct documents in the dataset}}$$

where a correct document is one that contains all the words in a query, and the selected cluster is that whose centroid has the highest average value for the query terms. That is, if a query comprises the words $\{w_1, w_2, \ldots, w_m\}$, and cluster $j$ has a centroid value $x_j^{w_i}$ for word $w_i$, then the cluster that best matches the query is the cluster $j^*$ such that

$$j^* = \arg\max_j \{\frac{1}{m}\sum_{i=1}^{m} x_j^{w_i}\}.$$

Our measure for retrieval performance considers only recall (*i.e.* how many correct documents were found for a particular query). It does not have a precision component, because the system retrieves only documents that contain all the words in the query. That is, precision is always perfect.

According to Figure 2, the recall for the 'hp' newsgroup is equally high with and without filtering when $k \geq 5$, as opposed to the other newsgroups, where the recall is significantly better with filtering for all values of $k$. We therefore chose $k = 5$ to evaluate retrieval, in order to expose the differences between the newsgroups.

Table 4 shows the retrieval performance obtained for the three queries, when clustering is performed with and without filtering, and with $k = 5$. The table shows the average performance of the pooled clusters separately for each of the three newsgroups. Also shown for each query is the total number of documents in the dataset that contain all the words in the query. The average performance of the best-matching cluster is displayed in bold font, and the standard deviation appears in brackets next to the performance.

The first query is related to the 'hp' newsgroup. The retrieval performance of the matching cluster

| filter | hp | tex | photoshop |
|---|---|---|---|
| Query 1: letter backend (total 25) | | | |
| off | **0.87** (0.32) | 0.07 (0.23) | 0.06 (0.22) |
| on | **0.93** (0.10) | 0.05 (0.10) | 0.02 (0.03) |
| Query 2: compile miktex (total 21) | | | |
| off | 0.20 (0.32) | **0.67** (0.36) | 0.13 (0.27) |
| on | 0.00 (0.01) | **0.99** (0.06) | 0.01 (0.06) |
| Query 3: rgb colour (total 22) | | | |
| off | 0.20 (0.21) | 0.11 (0.24) | **0.69** (0.30) |
| on | 0.15 (0.14) | 0.02 (0.12) | **0.83** (0.19) |

Table 4: Queries used to evaluate the retrieval task.

for this query is high with and without the filtering mechanism (the difference in performance is not statistically significant). As discussed above, this result is expected due to the similar recall score of the pooled cluster obtained with and without filtering for this newsgroup.

Filtering has a more significant effect for the queries relating to the other newsgroups. Query 2 is very specific to the 'tex' newsgroup: when filtering is used, almost all the relevant documents are retrieved by the corresponding cluster. The benefit of filtering is very clear when we consider the poor retrieval performance when filtering is not used: 33% of the documents are missed (the p-value for the difference in retrieval score is $\ll 0.01$). The third query has more ambiguity (the word 'colour' appears in the 'hp' newsgroup), and therefore the overall retrieval performance is worse than for the other queries. About 17% of the documents were missed when filtering was used, most of which were allocated to the 'hp' newsgroup. Nevertheless, the filtering mechanism has a significant effect even for this ambiguous query (p-value=0.03).

## 5 Conclusion

In this paper, we have identified features of electronic discussions that influence clustering performance, and presented a filtering mechanism that removes adverse influences. The effect of our filtering mechanism was evaluated by means of two experiments: coarse-level clustering and simple information retrieval. Our results show that filtering out the signature words of dominant speakers has a positive effect on clustering and retrieval performance. Although these experiments were performed at a coarser level of granularity than that of our target domain, our results indicate that filtering signature words is a promising pre-processing step for clustering electronic discussions.

From a more qualitative perspective, we clearly saw the benefit of the filtering mechanism in the example in Section 3.3 (Tables 2 and 3): when a generation component is used to describe the contents of clusters, the inclusion of author-specific words is uninformative and even confusing.

Our approach to filtering is general in the sense that we do not target specific parts of electronic discussions (*e.g.* the last few lines of a posting) for filtering. We have experimented with a more naive approach that removes all web and email addresses from a posting (they account for a significant portion of a signature). However, this simple heuristic yielded only a small improvement in clustering performance. More importantly, it clearly does not generalise to deal with the problem of identifying and removing author-specific terminology.

## 6 Acknowledgments

This research was supported in part by grant LP0347470 from the Australian Research Council and by an endowment from Hewlett Packard.

## References

Abdelmonem Abdelaziz Afifi and Virginia Ann Clark. 1996. *Computer-Aided Multivariate Analysis*. Chapman & Hall, London.

Yuval Marom and Ingrid Zukerman. 2004. Improving newsgroup clustering by filtering author-specific words. In *PRICAI'04 – Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*, Auckland, New Zealand.

J. L. Neto, A. D. Santos, C. A. A. Kaestner, and A. A. Freitas. 2000. Document clustering and text summarization. In *PAKDD-2000 – Proceedings of the 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining*, pages 41–55, London, UK.

G. Salton and M.J. McGill. 1983. *An Introduction to Modern Information Retrieval*. McGraw Hill.

Gerald Salton. 1971. Cluster search strategies and the optimization of retrieval effectiveness. In Gerald Salton, editor, *The SMART Retrieval System — Experiments in Automatic Document Processing*, pages 223–242. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Hinrich Schütze and Jan O. Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, Nevada.

Oren Zamir and Oren Etzioni. 1998. Web document clustering: A feasibility demonstration. In *SIGIR'98 – Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval*, pages 46–54, Melbourne, Australia.