WordNet-based Text Document Clustering

Julian Sedding
Department of Computer Science
University of York
Heslington, York YO10 5DD,
United Kingdom,
juliansedding@gmx.de

Dimitar Kazakov
AIG, Department of Computer Science
University of York
Heslington, York YO10 5DD,
United Kingdom,
kazakov@cs.york.ac.uk

Abstract

Text document clustering can greatly simplify browsing large collections of documents by reorganizing them into a smaller number of manageable clusters. Algorithms to solve this task exist; however, the algorithms are only as good as the data they work on. Problems include ambiguity and synonymy, the former allowing for erroneous groupings and the latter causing similarities between documents to go unnoticed. In this research, naïve, syntax-based disambiguation is attempted by assigning each word a part-of-speech tag and by enriching the 'bag-of-words' data representation often used for document clustering with synonyms and hypernyms from WordNet.

1 Introduction

Text document clustering is the grouping of text documents into semantically related groups, or as Hayes puts it, *"they are grouped because they are likely to be wanted together"*(Hayes, 1963). Initially, document clustering was developed to improve precision and recall of information retrieval systems. More recently, however, driven by the ever increasing amount of text documents available in corporate document repositories and on the Internet, the focus has shifted towards providing ways to efficiently browse large collections of documents and to reorganise search results for display in a structured, often hierarchical manner.

The clustering of Internet search results has attracted particular attention. Some recent studies explored the feasibility of clustering 'in real-time' and the problem of adequately labeling clusters. Zamir and Etzioni (1999) have created a clustering interface for the meta-search engine 'HuskySearch' and Zhang and Dong (2001) present their work on a system called SHOC. The reader is also referred to Vivisimo,[1] a commercial clustering interface based on results from a number of search-engines.

Ways to increase clustering speed are explored in many research papers, and the recent trend towards web-based clustering, requiring real-time performance, does not seem to change this. However, van Rijsbergen points out, *"it seems to me a little early in the day to insist on efficiency even before we know much about the behaviour of clustered files in terms of the effectiveness of retrieval"* (van Rijsbergen, 1989). Indeed, it may be worth exploring which factors influence the quality (or effectiveness) of document clustering.

Clustering can be broken down into two stages. The first one is to preprocess the documents, i.e. transforming the documents into a suitable and useful data representation. The second stage is to analyse the prepared data and divide it into clusters, i.e. the clustering algorithm.

Steinbach *et al.* (2000) compare the suitability of a number of algorithms for text clustering and conclude that bisecting k-means, a partitional algorithm, is the current state-of-the-art. Its processing time increases linearly with the number of documents and its quality is similar to that of hierarchical algorithms.

Preprocessing the documents is probably at least as important as the choice of an algorithm, since an algorithm can only be as good as the data it works on. While there are a number of preprocessing steps, that are almost standard now, the effects of adding background knowledge are still not very extensively researched. This work explores if and how the two following methods can improve the effectiveness of clustering.

[1] http://www.vivisimo.com

Part-of-Speech Tagging. Segond *et al.* (1997) observe that part-of-speech tagging (PoS) solves semantic ambiguity to some extent (40% in one of their tests). Based on this observation, we study whether naïve word sense disambiguation by PoS tagging can help to improve clustering results.

WordNet. Synonymy and hypernymy can reveal hidden similarities, potentially leading to better clusters. WordNet,[2] an ontology which models these two relations (among many others) (Miller et al., 1991), is used to include synonyms and hypernyms in the data representation and the effects on clustering quality are observed and analysed.

The overall aim of the approach outlined above is to cluster documents by meaning, hence it is relevant to language understanding. The approach has some of the characteristics expected from a robust language understanding system. Firstly, learning only relies on unannoted text data, which is abundant and does not contain the individual bias of an annotator. Secondly, the approach is based on general-purpose resources (Brill's PoS Tagger, WordNet), and the performance is studied under pessimistic (hence more realistic) assumptions, e.g., that the tagger is trained on a standard dataset with potentially different properties from the documents to be clustered. Similarly, the approach studies the potential benefits of using all possible senses (and hypernyms) from WordNet, in an attempt to postpone (or avoid altogether) the need for Word Sense Disambiguation (WSD), and the related pitfalls of a WSD tool which may be biased towards a specific domain or language style.

The remainder of the document is structured as follows. Section 2 describes related work and the techniques used to preprocess the data, as well as cluster it and evaluate the results achieved. Section 3 provides some background on the selected corpus, the Reuters-21578 test collection (Lewis, 1997b), and presents the subcorpora that are extracted for use in the experiments. Section 4 describes the experimental framework, while Section 5 presents the results and their evaluation. Finally, conclusions are drawn and further work discussed in Section 6.

[2]available at http://www.cogsci.princeton.edu/∼wn

2 Background

This work is most closely related to the recently published research of Hotho *et al.* (2003b), and can be seen as a logical continuation of their experiments. While these authors have analysed the benefits of using WordNet synonyms and up to five levels of hypernyms for document clustering (using the bisecting k-means algorithm), this work describes the impact of tagging the documents with PoS tags and/or adding all hypernyms to the information available for each document.

Here we use the vector space model, as described in the work of Salton *et al.* (1975), in which a document is represented as a vector or 'bag of words', i.e., by the words it contains and their frequency, regardless of their order. A number of fairly standard techniques have been used to preprocess the data. In addition, a combination of standard and custom software tools have been used to add PoS tags and WordNet categories to the data set. These will be described briefly below to allow for the experiments to be repeated.

The first preprocessing step is to PoS tag the corpus. The PoS tagger relies on the text structure and morphological differences to determine the appropriate part-of-speech. For this reason, if it is required, PoS tagging is the first step to be carried out. After this, stopword removal is performed, followed by stemming. This order is chosen to reduce the amount of words to be stemmed. The stemmed words are then looked up in WordNet and their corresponding synonyms and hypernyms are added to the bag-of-words. Once the document vectors are completed in this way, the frequency of each word across the corpus can be counted and every word occurring less often than the pre specified threshold is pruned. Finally, after the pruning step, the term weights are converted to $tfidf$ as described below.

Stemming, stopword removal and pruning all aim to improve clustering quality by removing noise, i.e. meaningless data. They all lead to a reduction in the number of dimensions in the term-space. Weighting is concerned with the estimation of the importance of individual terms. All of these have been used extensively and are considered the baseline for comparison in this work. However, the two techniques under investigation both add data to the representation. PoS tagging adds syntactic information and WordNet is used to add synonyms and hy-

pernyms. The rest of this section discusses preprocessing, clustering and evaluation in more detail.

PoS Tagging PoS tags are assigned to the corpus using Brill's PoS tagger. As PoS tagging requires the words to be in their original order this is done before any other modifications on the corpora.

Stopword Removal Stopwords, i.e. words thought not to convey any meaning, are removed from the text. The approach taken in this work does not compile a static list of stopwords, as usually done. Instead PoS information is exploited and all tokens that are not nouns, verbs or adjectives are removed.

Stemming Words with the same meaning appear in various morphological forms. To capture their similarity they are normalised into a common root-form, the stem. The morphology function provided with WordNet is used for stemming, because it only yields stems that are contained in the WordNet dictionary.

WordNet Categories WordNet, the lexical database developed by Miller *et al.*, is used to include background information on each word. Depending on the experiment setup, words are replaced with their synset IDs, which constitute their different possible senses, and also different levels of hypernyms, more general terms for the a word, are added.

Pruning Words appearing with low frequency throughout the corpus are unlikely to appear in more than a handful of documents and would therefore, even if they contributed any discriminating power, be likely to cause too fine grained distinctions for us to be useful, i.e clusters containing only one or two documents. Therefore all words (or synset IDs) that appear less often than a pre-specified threshold are pruned.

Weighting Weights are assigned to give an indication of the importance of a word. The most trivial weight is the word-frequency. However, more sophisticated methods can provide better results. Throughout this work, *tfidf* (term frequency x inverse document frequency) as described by Salton *et al.* (1975), is used.

One problem with term frequency is that the lengths of the documents are not taken into account. The straight forward solution to this problem is to divide the term frequency by the total number of terms in the document, the document length. Effectively, this approach is equivalent to normalising each document vector to length one and is called relative term frequency.

However, for this research a more sophisticated measure is used: the product of term frequency and inverse document frequency *tfidf*. Salton *et al.* define the *inverse document frequency idf* as

$$idf_t = \log_2 n - \log_2 df_t + 1 \qquad (1)$$

where df_t is the number of documents in which term t appears and n the total number of documents. Consequently, the *tfidf* measure is calculated as

$$tfidf_t = tf \cdot (\log_2 n - \log_2 df_t + 1) \qquad (2)$$

simply the multiplication of *tf* and *idf*. This means that larger weights are assigned to terms that appear relatively rarely throughout the corpus, but very frequently in individual documents. Salton *et al.* (1975) measure a 14% improvement in recall and precision for *tfidf* in comparison to the standard term frequency *tf*.

Clustering is done with the bisecting k-means algorithm as it is described by Steinbach *et al.* (2000). In their comparison of different algorithms they conclude that bisecting k-means is the current state of the art for document clustering. Bisecting k-means combines the strengths of partitional and hierarchical clustering methods by iteratively splitting the biggest cluster using the basic k-means algorithm. Basic k-means is a partitional clustering algorithm based on the vector space model. At the heart of such algorithms is a similarity measure. We choose the cosine distance, which measures the similarity of two documents by calculating the cosine of the angle between them. The cosine distance is defined as follows:

$$s(d_i, d_j) = \cos(\sphericalangle(\vec{d_i}, \vec{d_j})) = \frac{\vec{d_i} \cdot \vec{d_j}}{|\vec{d_i}| \cdot |\vec{d_j}|} \qquad (3)$$

where $|\vec{d_i}|$ and $|\vec{d_j}|$ are the lengths of vectors $\vec{d_i}$ and $\vec{d_j}$, respectively, and $\vec{d_i} \cdot \vec{d_j}$ is the dot-product of the two vectors. When the lengths of the vectors are normalised, the cosine distance is equivalent to the dot-product of the vectors, i.e. $\vec{d_1} \cdot \vec{d_2}$.

Evaluation Three different evaluation measures are used in this work, namely purity, entropy and overall similarity. Purity and entropy

are both based on precision,

$$\text{prec}(C, L) := \frac{|C \cap L|}{|C|}, \quad (4)$$

where each cluster C from a clustering $\mathbb{C}$ of the set of documents D is compared with the manually assigned category labels L from the manual categorisation $\mathbb{L}$, which requires a category-labeled corpus. Precision is the probability of a document in cluster C being labeled L.

Purity is the *percentage* of correctly clustered documents and can be calculated as:

$$\text{purity}(\mathbb{C}, \mathbb{L}) := \sum_{C \in \mathbb{C}} \frac{|C|}{|D|} \cdot \max_{L \in \mathbb{L}} \text{prec}(C, L) \quad (5)$$

yielding values in the range between 0 and 1.

The intra-cluster entropy (*ice*) of a cluster C, as described by Steinbach *et al.* (2000), considers the dispersion of documents in a cluster, and is defined as:

$$\text{ice}(C) := \sum_{L \in \mathbb{L}} \text{prec}(C, L) \cdot \log(\text{prec}(C, L)) \quad (6)$$

Based on the intra-cluster entropy of all clusters, the average, weighted by the cluster size, is calculated. This results in the following formula, which is based on the one used by Steinbach *et al.* (2000):

$$\text{entropy}(\mathbb{C}) := \sum_{C \in \mathbb{C}} \frac{|C|}{|D|} \cdot \text{ice}(C) \quad (7)$$

Overall similarity is independent of pre-annotation. Instead the intra-cluster similarities are calculated, giving an idea of the cohesiveness of a cluster. This is the average similarity between each pair of documents in a cluster, including the similarity of a document with itself. Steinbach *et al.* (2000) show that this is equivalent to the squared length of the cluster centroid, i.e. $|\vec{c}|^2$. The overall similarity is then calculated as

$$\text{overall similarity}(\mathbb{C}) := \sum_{C \in \mathbb{C}} \frac{|C|}{|D|} \cdot |\vec{c}|^2 \quad (8)$$

Similarity is expressed as a percentage, therefore the possible values for overall similarity range from 0 to 1.

3 The Corpus

Here we look at what kind of corpus is required to assess the quality of clusters, and present our choice, the Reuters-21578 test collection. This is followed by a discussion of the ways subcorpora can be extracted from the whole corpus in order to address some of the problems of the Reuters corpus.

A corpus useful for evaluating text document clustering needs to be annotated with class or category labels. This is not a straightforward task, as even human annotators sometimes disagree on which label to assign to a specific document. Therefore, all results depend on the quality of annotation. It is therefore unrealistic to aim at high rates of agreement with regard to the corpus, and any evaluation should rather focus on the relative comparison of the results achieved by different experiment setups and configurations.

Due to the aforementioned difficulty of agreeing on a categorisation and the lack of a definition of 'correct' classification, no standard corpora for evaluation of clustering techniques exist. Still, although not standardised, a number of pre-categorised corpora are available. Apart from various domain-specific corpora with class annotations, there is the Reuters-21578 test collection (Lewis, 1997b), which consists of 21578 newswire articles from 1987.

The Reuters corpus is chosen for use in the experiments of this projects for four reasons.

1. Its domain is not specific, therefore it can be understood by a non-expert.
2. WordNet, an ontology, which is not tailored to a specific domain, would not be effective for domains with a very specific vocabulary.
3. It is freely available for download.
4. It has been used in comparable studies before (Hotho et al., 2003b).

On closer inspection of the corpus, there remain some problems to solve. First of all, only about half of the documents are annotated with category-labels. On the other hand some documents are attributed to multiple categories, meaning that categories overlap. Some confusion seems to have been caused in the research community by the fact that there is a TOPICS attribute in the SGML, the value of which is either set to YES or NO (or BYPASS). However, this does not correspond to the values observed

within the TOPICS tag; sometimes categories can be found, even if the TOPICS attribute is set to NO and sometimes there are no categories assigned, even if the attribute indicates YES. Lewis explains that this is not an error in the corpus, but has to do with the evolution of the corpus and is kept for historic reasons (Lewis, 1997a).

Therefore, to prepare a base-corpus, the TOPICS attribute is ignored and all documents that have precisely one category assigned to them are selected. Additionally, all documents with an empty document body are also discarded. This results in the corpus 'reut-base' containing 9446 documents. The distribution of category sizes in the 'reut-base' is shown in Figure 1. It illustrates that there are a few categories occurring extremely often, in fact the two biggest categories contain about two thirds of all documents in the corpus. This unbalanced distribution would blur test results, because even 'random clustering' would potentially obtain purity values of 30% and more only due to the contribution of the two main categories.

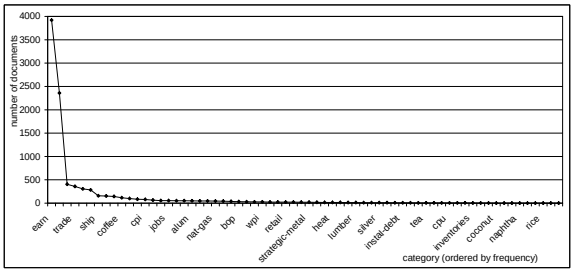


Figure 1: Category Distribution for Corpus 'reut-base' (only selected categories are listed).

Similar to Hotho *et al.* (2003b), we get around this problem by deriving new corpora from the base corpus. Their maximum category size is reduced to 20, 50 and 100 documents respectively. Categories containing more documents are not excluded, but instead they are reduced in size to comply with the defined maximum, i.e., all documents in excess of the maximum are removed.

Creating derived corpora has the further advantages of reducing the size of corpora and thus computational requirements for the test runs. Also, tests can be run on more and less homogeneous corpora, homogeneous with regard to the cluster size, that is, which can give an idea of how the method performs under different conditions. Especially for this purpose a fourth, extremely homogeneous test corpus, 'reut-min15-max20' is derived. It is like the 'reut-max20' corpus, but all categories containing less than 15 documents are entirely removed. The 'reut-min15-max20' is thus the most homogeneous test corpus, with a standard deviation in cluster size of only 0.7 documents.

A summary of the derived test corpora is shown in Table 1, including the number of documents they contain, i.e. their size, the average category size and the standard deviation. Figure 2 shows the distribution of categories within the derived corpora graphically.

Table 1: Corpus Statistics

Name	Size	Category Size ø	Category Size stdev
reut-min15-max20	713	20	0.7
reut-max20	881	13	7.7
reut-max50	1690	24	19.9
reut-max100	2244	34	35.2
reut-base	9446	143	553.2

Note: The average category size is rounded to the nearest whole number, the standard deviation to the first decimal place.

4 Clustering with PoS and Background Knowledge

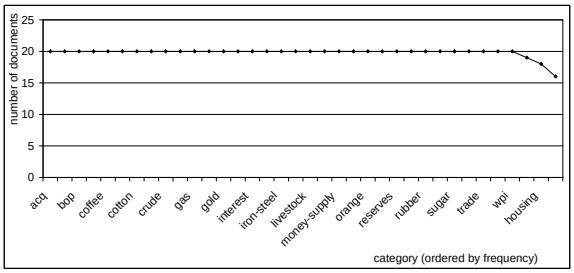
The aim of this work is to explore the benefits of partial disambiguation of words by their PoS and the inclusion of WordNet concepts. This has been tested on five different setups, as shown in Table 2.

Table 2: Experiment Configurations

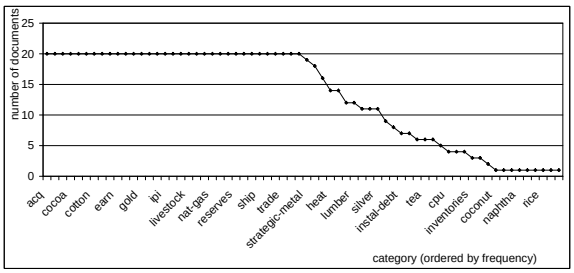
Name	**Base**	**PoS**	**Syns**	**Hyper**
Baseline	yes			
PoS_Only	yes	yes		
Syns	yes	yes	yes	
Hyper_5	yes	yes	yes	5
Hyper_All	yes	yes	yes	all

Base: stopword removal, stemming, pruning and *tf idf* weighting are performed; PoS tags are stripped.
PoS: PoS tags are kept attached to the words.
Syns: all senses of a word are included using synset offsets.
Hyper: hypernyms to the specified depth are included.
Empty fields indicate 'no' or '0'.

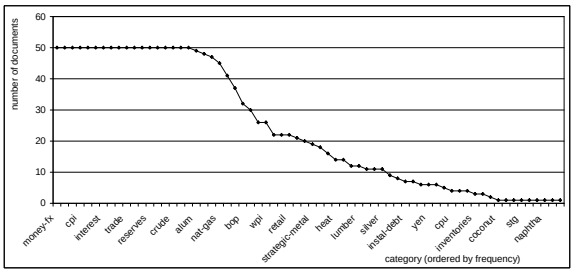
Baseline The first configuration setting is used to get a baseline for comparison. All basic preprocessing techniques are used, i.e. stopword removal, stemming, pruning and



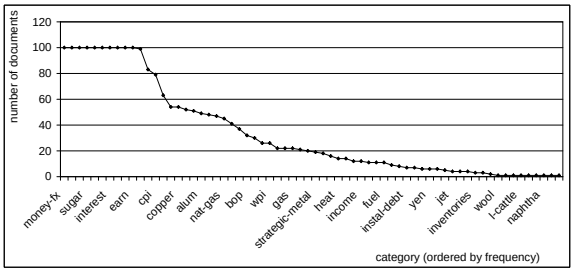
reut-min15-max20



reut-max20



reut-max50



reut-max100

Figure 2: Category Distributions for Derived Corpora

weighting. PoS tags are removed from the tokens to get the equivalent of a raw text corpus.

PoS_Only Identical to the baseline, but the PoS tags are not removed.

Syns In addition to the previous configuration, all WordNet senses (synset IDs) of each PoS tagged token are included.

Hyper_5 Here five levels of hypernyms are included in addition to the synset IDs.

Hyper_All Same as above, but all hypernyms for each word token are included.

Each of the configurations is used to create 16, 32 and 64 clusters from each of the four test-corpora. Due to the random choice of initial cluster centroids in the bisecting k-means algorithm, the means of three test-runs with the same configuration is calculated for corpora 'reut-max20' and 'reut-max50'. The existing project time constraints allowed us to gain some additional insight by doing one test-run for each of 'reut-max100' and 'reut-min15-max20'. This results in 120 experiments in total.

All configurations use $tfidf$ weighting and pruning. The pruning thresholds vary. For all experiments using the 'reut-max20' corpus all terms occurring less than 20 times are pruned. The experiments on corpora 'reut-max50' and 'reut-min15-max20' are carried out with a pruning threshold of 50. For the corpus 'reut-max100', the pruning threshold is set to 50 when configurations **Baseline**, **PoS_Only** or **Syns** are used and to 200 otherwise. This relatively high threshold is chosen, in order to reduce memory requirements. To ensure that this inconsistency does not distort the conclusions drawn from the test data, the results of these tests are considered with great care and are explicitly referred to when used.

Further details of this research are described in an unpublished report (Sedding, 2004).

5 Results and Evaluation

The results are presented in the format of one graph per corpus, showing the entropy, purity and overall similarity values for each of the configurations shown in Table 2.

On the X-axis, the different configuration settings are listed. On the right-hand side, *hype* refers to the hypernym depth, *syn* refers to whether synonyms were included or not, *pos* refers to the presence or absence of PoS tags and *clusters* refers to the number of clusters created. For improved readability, lines are drawn, splitting the graphs into three sections, one for each number of clusters. For experiments on the corpora 'reut-max20' and 'reut-max50', the values in the graphs are the average of three test runs, whereas for the corpora 'reut-min15-

max20' and 'reut-max100', the values are those obtained from a single test run.

The Y-axis indicates the numerical values for each of the measures. Note that the values for purity and similarity are percentages, and thus limited to the range between 0 and 1. For those two measures, higher values indicate better quality. High entropy values, on the other hand, indicate lower quality. Entropy values are always greater than 0 and for the particular experiments carried out, they never exceed 1.3.

In analysing the test results, the main focus is on the data of corpora 'reut-max20' and 'reut-max50', shown in Figure 3 and Figure 4, respectively. This data is more reliable, because it is the average of repeated test runs. Figures 6–7 show the test data obtained from clustering the corpora 'reut-min15-max20' and 'reut-max100', respectively.

The fact that the purity and similarity values are far from 100 percent is not unusual. In many cases, not even human annotators agree on how to categorise a particular document (Hotho et al., 2003a). More importantly, the number of categories are not adjusted to the number of labels present in a corpus, which makes complete agreement impossible.

All three measures indicate that the quality increases with the number of clusters. The graph in Figure 5 illustrates this for the entropy in 'reut-max50'. For any given configuration, it appears that the decrease in entropy is almost constant when the number of clusters increases. This is easily explained by the average cluster sizes, which decrease with an increasing number of clusters; when clusters are smaller, the probability of having a high percentage of documents with the same label in a cluster increases. This becomes obvious when very small clusters are looked at. For instance, the minimum purity value for a cluster containing three documents is 33 percent, for two documents it is 50 percent, and, in the extreme case of a single document per cluster, purity is always 100 percent.

The **PoS_Only** experiment results in performance, which is very similar to the **Baseline**, and is sometimes a little better, sometimes a little worse. This is expected, and the experiment is included to allow for a more accurate interpretation of the subsequent experiments using synonyms and hypernyms.

A more interesting observation is that purity and entropy values indicate better clusters for **Baseline** than for any of the configurations using background knowledge from WordNet (i.e. **Syns**, **Hyper_5** and **Hyper_All**). One possible conclusion is that adding background knowledge is not helpful at all. However, the reasons for the relatively poor performance could also be due to the way the experiments are set up. Therefore, a possible explanation for these results could be that the benefit of extra overlap between documents, which the added synonyms and hypernyms should provide, is outweighed by the additional noise they create. WordNet does often provide five or more senses for a word, which means that for one correct sense a number of incorrect senses are added, even if the PoS tags eliminate some of them.

The overall similarity measure gives a different indication. Its values appear to increase for the cases where background knowledge is included, especially when hypernyms are added. Overall similarity is the weighted average of the intra-cluster similarities of all clusters. So the intra-cluster similarity actually increases with

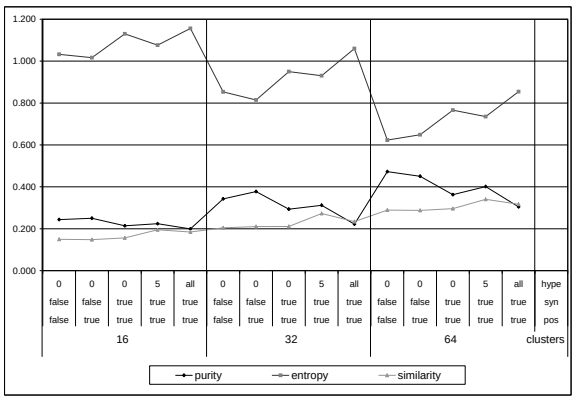


Figure 3: Test Results for 'reut-max20'

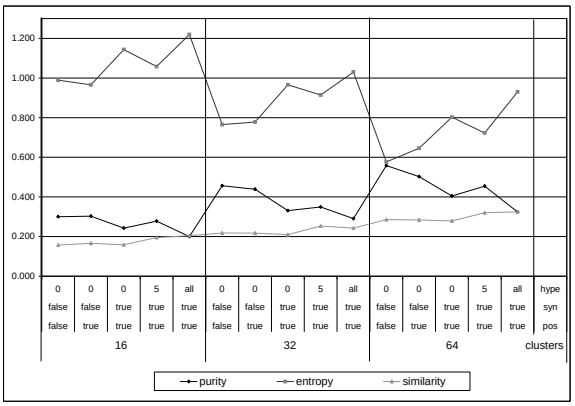


Figure 4: Test Results for 'reut-max50'

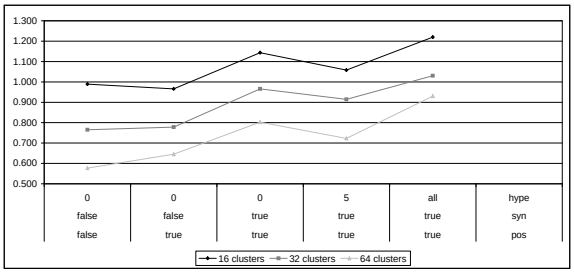


Figure 5: Entropies for Different Cluster Sizes in 'reut-max50'

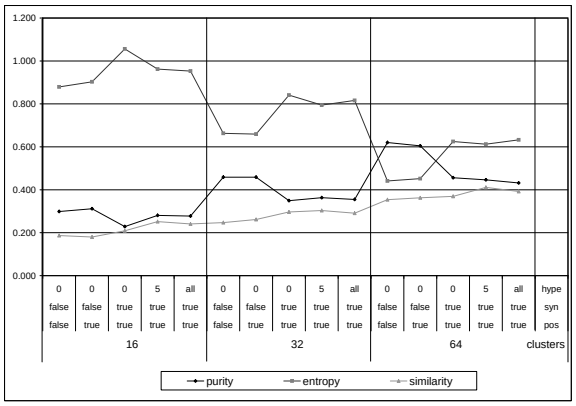


Figure 6: Test Results for 'reut-min15-max20'

added information. As similarity increases with additional overlap, the overall similarity measure shows that additional overlap is achieved.

The main problem with the approach of adding *all* synonyms and *all* hypernyms into the document vectors seems to be the added noise. The expectation that $tfidf$ weighting would take care of these quasi-random new concepts is not met, but the results also indicate possible improvements to this approach.

If word-by-word disambiguation would be

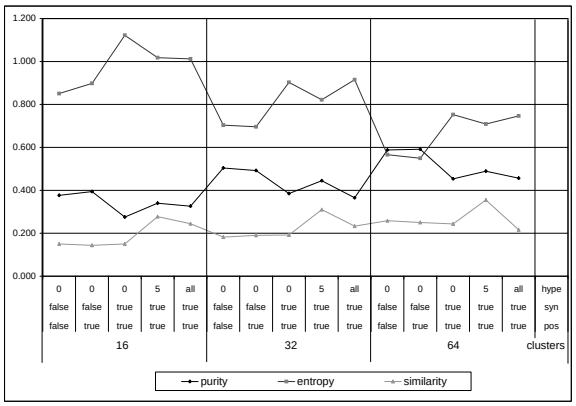


Figure 7: Test Results for 'reut-max100'

used, the correct sense of a word could be chosen and only the hypernyms for the correct sense of the word could be taken into account. This should drastically reduce noise. The benefit of the added 'correct' concepts would then probably improve cluster quality. Hotho *et al.* (2003a) experimented successfully with simple disambiguation strategies, e.g., they used only the first sense provided by WordNet.

As an alternative to word-by-word disambiguation, a strategy to disambiguate based on document vectors could be devised; after adding all alternative senses of the terms, the least frequent ones could be removed. This is similar to pruning but would be done on a document by document basis, rather than globally on the whole corpus. The basis for this idea is that only concepts that appear repeatedly in a document contribute (significantly) to the meaning of the document. It is important that this is done before hypernyms are added, especially when all levels of hypernyms are added, because the most general terms are bound to appear more often than the more specific ones. This would lead to lots of very similar, but meaningless bags of words or bags of concepts.

Comparing **Syns**, **Hyper_5** and **Hyper_All** with each other, in many cases **Hyper_5** gives the best results. A possible explanation could again be the equilibrium between valuable information and noise that are added to the vector representations. From these results it seems that there is a point where the amount of information added reaches its maximum benefit; adding more knowledge afterwards results in decreased cluster quality again. It should be noted that a fixed threshold for the levels of hypernyms used is unlikely to be optimal for all words. Instead, a more refined approach could set this threshold as a function of the semantic distance (Resnik and Yarowsky, 2000; Stetina, 1997) between the word and its hypernyms.

The maximised benefit is most evident in the 'reut-max100' corpus (Figure 7). However, it needs to be kept in mind that for the last two data points, **Hyper_5** and **Hyper_All**, the pruning threshold is 200. Therefore, the comparison with **Syns** needs to be done with care. This is not much of a problem, because the other graphs consistently show that the performance for **Syns** is worse than for **Hyper_5**. The difference between **Hyper_5** and **Hyper_All** in 'reut-max100', can be directly compared though, because the pruning thresh-

old of 200 is used for both configurations.

Surprisingly, there is a sharp drop in the overall similarity from **Hyper_5** to **Hyper_All**, much more evident than in the other three corpora. One possible explanation could be the different structure of the corpus. It seems more probable, however, that the high pruning threshold is the cause again. Assuming that **Hyper_5** seldom includes the most general concepts, whereas **Hyper_All** always includes them, their frequency in **Hyper_All** becomes so high that the frequencies of all the other terms are very low in comparison. The document vectors in case of **Hyper_All** end up containing mostly meaningless concepts, because most of the others are pruned. This leads to decreased cluster quality because the general concepts have little discriminating power. In the corresponding experiments on other corpora, more of the specific concepts are retained. Therefore, a better balance between general and specific concepts is maintained, keeping the cluster quality higher than in the case of corpus 'reut-max100'.

PoS_Only performs similar to **Baseline**, although usually a slight decrease in quality can be observed. Despite the assumption that the disambiguation achieved by the PoS tags should improve clustering results, this is clearly not the case. PoS tags only disambiguate the cases where different word classes are represented by the same stem, e.g., the noun 'run' and the verb 'run'. Clearly the meanings of these pairs are in most cases related. Therefore, distinguishing between them reduces the weight of their common concept by splitting it between two concepts. In the worst case, they are pruned if treated separately, instead of contributing significantly to the document vector as a joint concept.

6 Conclusions

The main finding of this work is that including synonyms and hypernyms, disambiguated only by PoS tags, is not successful in improving clustering effectiveness. This could be attributed to the noise introduced by all incorrect senses that are retrieved from WordNet. It appears that disambiguation by PoS alone is insufficient to reveal the full potential of including background knowledge. One obviously impractical alternative would be manual sense disambiguation. The automated approach of only using the most common sense adopted by Hotho *et al.*(2003b) seems more realistic yet beneficial.

When comparing the use of different levels of hypernyms, the results indicate that including only five levels is better than including all. A possible explanation of this is that the terms become too general when all hypernym levels are included.

Further research is needed to determine whether this way of document clustering can be improved by appropriately selecting a subset of the synonyms and hypernyms used here. There is a number of corpus-based approaches to word-sense disambiguation (Resnik and Yarowsky, 2000), which could be used for this purpose.

The other point of interest that could be further analysed is to find out why using five levels of hypernyms produces better results than using all levels of hypernyms. It would be interesting to see whether this effect persists when better disambiguation is used to determine 'correct' word senses.

Acknowledgements

The authors wish to thank the three anonymous referees for their valuable comments.

References

R.M. Hayes. 1963. Mathematical models in information retrieval. In P.L. Garvin, editor, *Natural Language and the Computer*, page 287. McGraw-Hill, New York.

A. Hotho, S. Staab, and G. Stumme. 2003a. Text clustering based on background knowledge. *Technical Report No. 425.*

A. Hotho, S. Staab, and G. Stumme. 2003b. Wordnet improves text document clustering. *In Proceedings of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIR Conference.*

D.D. Lewis. 1997a. Readme file of Reuters-21578 text categorization test collection, distribution 1.0.

D.D. Lewis. 1997b. Reuters-21578 text categorization test collection, distribution 1.0.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1991. Five papers on wordnet. *International Journal of Lexicography.*

P. Resnik and D. Yarowsky. 2000. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133.

G. Salton, A. Wong, and C.S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18:613–620.

J. Sedding. 2004. Wordnet-based text document clustering. Bachelor's Thesis.

F. Segond, A. Schiller, G. Grefenstette, and J.P. Chanod. 1997. An experiment in semantic tagging using hidden Markov model tagging. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 78–81. Association for Computational Linguistics, New Brunswick, New Jersey.

M. Steinbach, G. Karypis, and V. Kumar. 2000. A comparison of document clustering techniques. *In KDD Workshop on Text Mining.*

Jiri Stetina. 1997. *Corpus Based Natural Language Ambiguity Resolution.* Ph.D. thesis, Kyoto University.

C.J. van Rijsbergen. 1989. *Information Retrieval (Second Edition).* Buttersworth, London.

O. Zamir and O. Etzioni. 1999. Grouper: A dynamic clustering interface to Web search results. *Computer Networks, Amsterdam, Netherlands*, 31(11–16):1361–1374.

D. Zhang and Y. Dong. 2001. Semantic, hierarchical, online clustering of web search results. *3rd International Workshop on Web Information and Data Management, Atlanta, Georgia.*