Wordnet improves Text Document Clustering

Andreas Hotho HOTHO@AIFB.UNI-KARLSRUHE.DE
Steffen Staab STAAB@AIFB.UNI-KARLSRUHE.DE
Gerd Stumme STUMME@AIFB.UNI-KARLSRUHE.DE
Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany

Abstract

Text document clustering plays an important role in providing intuitive navigation and browsing mechanisms by organizing large amounts of information into a small number of meaningful clusters. The bag of words representation used for these clustering methods is often unsatisfactory as it ignores relationships between important terms that do not co-occur literally. In order to deal with the problem, we integrate background knowledge — in our application Wordnet — into the process of clustering text documents. We cluster the documents by a standard partitional algorithm. Our experimental evaluation on Reuters newsfeeds compares clustering results with pre-categorizations of news. In the experiments, improvements of results by background knowledge compared to the baseline can be shown for many interesting tasks.

1. Introduction

With the abundance of text documents available through corporate document management systems and the World Wide Web, the efficient, high-quality partitioning of texts into previously unseen categories is a major topic for applications such as information retrieval from databases, business intelligence solutions or enterprise portals. So far, however, existing text clustering solutions only relate documents that use identical terminology, while they ignore *conceptual similarity* of terms such as defined in terminological resources like WordNet (Miller, 1995).

In this paper we investigate which beneficial effects can be achieved for text document clustering by integrating an explicit conceptual account of terms found in WordNet. In order to come up with this result we have performed an empirical evaluation. We compare a simple baseline (Section 2) with different strategies for representing text documents that take background knowledge into account to various extent (Section 3). For instance, terms like "beef" and "pork" are found to be similar, because they both are subconcepts of "meat" in WordNet. The clustering is then performed with Bi-Section-KMeans, which has been shown to perform as good as other text clustering algorithms — and frequently better (cf. the seminal paper (Steinbach et al., 2000)). For the evaluation (cf. Section 4), we have investigated the Reuters corpus on newsfeeds, which comes with a set of categorizing labels attached to the documents. The evaluation results (cf. Section 5) compare the original classification with the partitioning produced by clustering the different representations of the text documents. Furthermore, by analysing the manually defined Reuters categories, we find explanations of when background knowledge helps.

In Section 6, we point to some related work. Finally, we conclude that the best strategies that involve background knowledge are most often better than the baseline when word sense disambiguation and feature weighting are included (Section 7).

2. Baseline Text Document Representation

For the clustering experiments described subsequently, we have prepared different representations of text documents suitable for the clustering algorithms.

Let us first consider documents to be bags of terms (cf. (Salton, 1989)). Let $\text{tf}(d,t)$ be the absolute frequency of term $t \in T$ in document $d \in D$, where D is the set of documents and $T = \{t_1, \ldots, t_m\}$ is the set all different terms occurring in D. We denote the term vectors $\vec{t_d} = (\text{tf}(d,t_1), \ldots, \text{tf}(d,t_m))$. Later on, we will need the notion of the centroid of a set X of term vectors. It is defined as the mean value $\vec{t_X} := \frac{1}{|X|}\sum_{\vec{t_d}\in X}\vec{t_d}$ of its term vectors. In the sequel, we will apply tf also on sets of terms: for $T' \subseteq T$, we let $\text{tf}(d,T') := \sum_{t\in T'}\text{tf}(d,t)$.

As initial approach we have produced this standard representation of the texts by term vectors. The initial term vectors are further modified as follows.

Stopwords are words which are considered as non–

descriptive within a bag–of–words approach. Following common practice, we removed stopwords from T, using a standard list with 571 stopwords.[1]

We have processed our text documents using the Porter stemmer introduced in (Porter, 1980). We used the *stemmed terms* to construct a vector representation $\vec{t_d}$ for each text document.

Then, we have investigated how pruning rare terms affects results. Depending on a pre-defined threshold δ, a term t is discarded from the representation (i. e., from the set T), if $\sum_{d \in D} \text{tf}(d,t) \leq \delta$. We have used the values 0, 5 and 30 for δ. The rationale behind *pruning* is that infrequent terms do not help for identifying appropriate clusters, but may still add noise to the distance measures degrading overall performance.[2]

Tfidf weighs the frequency of a term in a document with a factor that discounts its importance when it appears in almost all documents. The tfidf (term frequency$-$inverted document frequency)[3] of term t in document d is defined by:

$$\text{tfidf}(d,t) := \log(\text{tf}(d,t)+1) * \log\left(\frac{|D|}{\text{df}(t)}\right)$$

where $\text{df}(t)$ is the document frequency of term t that counts in how many documents term t appears. If tfidf weighting is applied then we replace the term vectors $\vec{t_d} := (\text{tf}(d,t_1), \ldots, \text{tf}(d,t_m))$ by $\vec{t_d} := (\text{tfidf}(d,t_1), \ldots, \text{tfidf}(d,t_m))$. There are more sophisticated measures than tfidf in the literature (see, e. g., (Amati et al., 2001)), but we abstract herefrom, as this is not the main topic of this paper.

Based on the initial text document representation, we have first applied stopword removal. Then we performed stemming, pruning and tfidf weighting in all different combinations. This also holds for the initial document representation involving background knowledge described subsequently. When stemming and/or pruning and/or tfidf weighting was performed, we have always performed them in the order in which they have been listed here.

3. Compiling Background Knowledge into the Text Document Representation

The background knowledge we have exploited is given through a simple ontology. We first describe its structure, then the actual ontology and its integration into the initial text document representation by various strategies.

3.1. Ontology

The background knowledge we will exploit further on is encoded in a *core ontology*. We here present those parts of our wider ontology definition (cf. (Bozsak et al., 2002)) that we have exploited:

Definition: A *core ontology* is a tuple $\mathcal{O} := (C, \leq_C)$ consisting of a set C whose elements are called *concept identifiers*, and a partial order $\leq_C$ on C, called *concept hierarchy* or *taxonomy*.

Often we will call concept identifiers just *concepts*, for sake of simplicity.

Definition: If $c_1 <_C c_2$, for $c_1, c_2 \in C$, then c_1 is a *subconcept of* c_2, and c_2 is a *superconcept of* c_1. If $c_1 <_C c_2$ and there is no $c_3 \in C$ with $c_1 <_C c_3 <_C c_2$, then c_1 is a *direct subconcept of* c_2, and c_2 is a *direct superconcept of* c_1. We note this by $c_1 \prec c_2$.

According to the international standard ISO 704, we provide names for the concepts (and relations). Instead of 'name', we here call them 'sign' or 'lexical entries' to better describe the functions for which they are used.

Definition: A *lexicon* for an ontology $\mathcal{O}$ is a tuple $Lex := (S_C, Ref_C)$ consisting of a set S_C whose elements are called *signs for concepts*, and a relation $Ref_C \subseteq S_C \times C$ called *lexical reference for concepts*, where $(c,c) \in Ref_C$ holds for all $c \in C \cap S_C$. Based on Ref_C, we define, for $s \in S_C$, $Ref_C(s) := \{c \in C \mid (s,c) \in Ref_C\}$ and, for $c \in C$, $Ref_C^{-1}(c) := \{s \in S_C \mid (s,c) \in Ref_C\}$. An *ontology with lexicon* is a pair $(\mathcal{O}, Lex)$ where $\mathcal{O}$ is an ontology and Lex is a lexicon for $\mathcal{O}$.

This definition allows for a very generic approach towards using ontologies for clustering. For the purpose of actual evaluation of clustering with background knowledge, we needed a specific resource, which fits to the document collection. We have chosen Wordnet 1.7,[4] as it fits to the generality of the Reuters corpus. Wordnet (Miller, 1995) comprises a core ontology and a lexicon. It consists of 109377 concepts (synsets in Wordnet terminology) and 144684 lexical entries[5] (called words in Wordnet). One example synset is "foot, ft" and a corresponding word is "foot". In Wordnet, the function Ref_C relates terms if they have a lexical entry (e.g., $s_1 =$ "foot" and $s_2 =$ "feet") with their corresponding concepts (e.g., synsets $c_1 =$ "foot, ft", $c_2 =$ "foot, human foot, pes", ...). Thus, for a term t appearing in a document d, $Ref_C(t)$ allows for retrieving its corre-

[1] See http://www.aifb.uni-karlsruhe.de/WBS/aho/clustering

[2] We have also investigated the influence of the document frequency of a term t for pruning, but it showed that this parameter hardly effects the clustering results.

[3] tfidf actually refers to a class of weighting schemata. Above we have given the one we have used.

[4] http://www.cogsci.princeton.edu/˜wn/obtain.shtml

[5] The actual number of lexical entries is higher in our count, as for one stem like "foot", Wordnet includes several morphological derivations like "feet".

sponding concepts.

In addition, Wordnet provides a ranking on the set $\mathit{Ref}_C(s)$ for each lexical entry s indicating the frequency of its usage in English language. For example, $\mathit{Ref}_C(s_1)$ returns as the first concept c_1 and then c_2. Corresponding to our definition of a core ontology, Wordnet also offers access functions to its concept hierarchy $\leq_C$.

So far, from all the descriptions given in Wordnet, we have exploited only information about nouns. I.e., we have used only 68.1% of the synsets available in Wordnet.

Using the morphological capabilities of Wordnet rather than a Porter stemmer we achieved improved results. Therefore, when using background knowledge, stemming has only been performed for terms that do not appear as lexical entries in Wordnet.

3.2. Term vs. Concepts Vector Strategies

Enriching the term vectors with concepts from the core ontology has two benefits. First it resolves synonyms; and second it introduces more general concepts which help identifying related topics. For instance, a document about beef may not be related to a document about pork by the cluster algorithm if there are only 'beef' and 'pork' in the term vector. But if the more general concept 'meat' is added to both documents, their semantical relationship is revealed. We have investigated different strategies for adding or replacing terms by concepts:

Add Concepts ("add"[6]). When applying this strategy, we have extended each term vector $\vec{t_d}$ by new entries for Wordnet concepts c appearing in the document set. Thus, the vector $\vec{t_d}$ was replaced by the concatenation of $\vec{t_d}$ and $\vec{c_d}$, where $\vec{c_d} := (\mathrm{cf}(d, c_1), \ldots, \mathrm{cf}(d, c_l))$ is the concept vector with $l = |C|$ and $\mathrm{cf}(d, c)$ denotes the frequency that a concept $c \in C$ appears in a document d as indicated by applying the reference function Ref_C to all terms in the document d. For a detailed definition of cf, see next subsection.
Hence, a term that also appeared in Wordnet as a synset would be accounted for at least twice in the new vector representation, i. e., once as part of the old $\vec{t_d}$ and at least once as part of $\vec{c_d}$. It could be accounted for also more often, because a term like "bank" has several corresponding concepts in Wordnet.

Replace Terms by Concepts ("repl"). This strategy works like 'Add Concepts' but it expels all terms from the vector representations $\vec{t_d}$ for which at least one corresponding concept exists. Thus, terms that appear in Wordnet are only accounted at the concept level, but terms that do not appear in Wordnet are not discarded.

Concept Vector Only ("only"). This strategy works like 'Replace Terms by Concepts' but it expels *all* terms from the vector representation. Thus, terms that do not appear in Wordnet are discarded; $\vec{c_d}$ is used to represent document d.

3.3. Strategies for Disambiguation

The assignment of terms to concepts in Wordnet is ambiguous. Therefore, adding or replacing terms by concepts may add noise to the representation and may induce a loss of information. Therefore, we have also investigated how the choice of a "most appropriate" concept from the set of alternatives may influence the clustering results.

While there is a whole field of research dedicated to word sense disambiguation (e.g., cf. (Ide & Véronis, 1998)), it has not been our intention to determine which one could be the most appropriate, but simply whether word sense disambiguation is needed at all. For this purpose, we have considered two simple disambiguation strategies besides of the baseline:

All Concepts ("all"). The baseline strategy is not to do anything about disambiguation and consider all concepts for augmenting the text document representation. Then, the concept frequencies are calculated as follows:

$$\mathrm{cf}(d, c) := \mathrm{tf}(d, \{t \in T \mid c \in \mathit{Ref}_C(t)\}) \ .$$

First Concept ("first"). As mentioned in Sec. 3.1, Wordnet returns an *ordered* list of concepts when applying Ref_C to a set of terms. Thereby, the ordering is supposed to reflect how common it is that a term reflects a concept in "standard" English language. More common term meanings are listed before less common ones.
For a term t appearing in S_C, this strategy counts only the concept frequency cf for the first ranked element of $\mathit{Ref}_C(t)$, i.e. the most common meaning of t. For the other elements of $\mathit{Ref}_C(t)$, frequencies of concepts are not increased by the occurrence of t. Thus the concept frequency is calculated by:

$$\mathrm{cf}(d, c) := \mathrm{tf}(d, \{t \in T \mid \mathrm{first}(\mathit{Ref}_C(t)) = c\})$$

where $\mathrm{first}(\mathit{Ref}_C)$ gives the first concept $c \in \mathit{Ref}_C$ according to the order from Wordnet.

Disambiguation by Context ("context"). The sense of a term t that refers to several different concepts $\mathit{Ref}_C(t) := \{b, c, \ldots\}$ may be disambiguated by a simplified version of (Agirre & Rigau, 1996)'s strategy:

1. Define the semantic vicinity of a concept c to be the set of all its direct sub- and superconcepts
$$V(c) := \{b \in C | c \prec b \text{ or } b \prec c\}.$$
2. Collect all terms that could express a concept from the conceptual vicinity of c by
$$U(c) := \bigcup_{b \in V(c)} \mathit{Ref}_C^{-1}(b).$$
3. The function dis: $D \times T \rightarrow C$ with $\mathrm{dis}(d, t) :=$

[6] These abbreviations are used below in Section 5.2

$\text{first}\{c \in \mathit{Ref}_C(t) \mid c \text{ maximizes } \text{tf}(d, U(c))\}$ disambiguates term t based on the context provided by document d.

4. Let $\text{cf}(d, c) := \text{tf}(d, \{t \in T \mid \text{dis}(d, t) = c\})$.

3.4. Strategies for considering Hypernyms

The third set of strategies varies the amount of background knowledge. Its principal idea is that if a term like 'beef' appears, one does not only represent the document by the concept corresponding to 'beef', but also by the concepts corresponding to 'meat' and 'food' etc. up to a certain level of generality. The following procedure realizes this idea by adding to the concept frequency of higher level concepts in a document d the frequencies that their subconcepts (at most r levels down in the hierarchy) appear, *i.e.* for $r \in \mathbb{N}_0$: The vectors we consider are of the form

$$\vec{t_d} := (\text{tf}(d, t_1), \ldots, \text{tf}(d, t_m), \text{cf}(d, c_1), \ldots, \text{cf}(d, c_n))$$

(the concatenation of an initial term representation with a concept vector). Then the frequencies of the concept vector part are updated in the following way: For all $c \in C$, replace $\text{cf}(d, c)$ by

$$\text{cf}'(d, c) := \sum_{b \in H(c,r)} \text{cf}(d, b),$$

where $H(c,r) := \{c' | \exists c_1, \ldots, c_i \in C\colon c' \prec c_1 \prec \ldots \prec c_i = c,\ 0 \leq i \leq r\}$ gives for a given concept c the r next subconceps in the taxonomy. In particular $H(c, \infty)$ returns all subconcepts of c. This implies: The strategy $r = 0$ does not change the given concept frequencies, $r = n$ adds to each concept the frequency counts of all subconcepts in the n levels below it in the ontology and $r = \infty$ adds to each concept the frequency counts of all its subconcepts.

4. Partitional Clustering

Our incorporation of background knowledge is rather independent of the concrete clustering method. The only requirements we had were that the baseline could achieve good clustering results in an efficient way on the Reuters corpus. In (Steinbach et al., 2000) it has been shown that Bi-Section-KMeans – a variant of KMeans – fulfilled these conditions, while frequently outperforming standard KMeans as well as agglomerative clustering techniques.

For our experiments, the similarity between two text documents $d_1, d_2 \in D$ is measured by the cosine of the angle between the vectors $\vec{t}_1, \vec{t}_2$ representing them:

$$\cos(\sphericalangle(\vec{t}_1, \vec{t}_2)) = \frac{\vec{t}_1 \cdot \vec{t}_2}{\|\vec{t}_1\| \cdot \|\vec{t}_2\|}$$

4.1. Evaluation Setting

The principal idea of the experiments was the comparison of clustering results on a standard text corpus against a manually predefined categorization of the corpus. Such a predefined categorization exists only for few text corpora. We have chosen the Reuters-21578 news corpus ((Lewis, 1997)[7], cf. section 4.3), because it comprises an *a priori* categorization of documents, its domain is broad enough to be realistic, and the content of the news were understandable for non-experts (like us) in order to be able to explain results. Furthermore, Reuters-21578 is a well-known, freely available and well investigated corpus.

Important reasons for us to use Wordnet as a core ontology in conjunction with Reuters-21578 as a corpus were that Wordnet is freely available and that it has not been specifically designed to facilitate the clustering task. We performed a second evaluation on the FAO Document Online Catalogue,[8] in which the Food and Agriculture Organization (FAO) of the United Nations stores documents about agriculture, which are labeled with the controlled vocabulary AGROVOC.[9] The evaluation on this domain and with this specific ontology provided similar results, which we omit here because of space restrictions.

In the experiments we have varied the different strategies for plain term vector representation and for vector representations containing background knowledge as elaborated in Sections 2 and 3. We have clustered the representations using Bi-Section-KMeans and have compared the pre-categorization with our clustering results using standard measures for this task, as defined below.

4.2. Evaluation Measures

The purity measure is based on the precision measure as well-known from information retrieval (cf. (Pantel & Lin, 2002)). Each resulting cluster P from a partitioning $\mathbb{P}$ of the overall document set D is treated as if it were the result of a query. Each set L of documents of a partitioning $\mathbb{L}$ which is obtained by manually labeling is treated as if it were the desired set of documents for a query. The two partitionings $\mathbb{P}$ and $\mathbb{L}$ are then compared as follows.

The precision of a cluster $P \in \mathbb{P}$ for a given category $L \in \mathbb{L}$ is given by $\text{Precision}(P, L) := \frac{|P \cap L|}{|P|}$. The overall value for purity is computed by taking the weighted average of maximal precision values:

$$\text{Purity}(\mathbb{P}, \mathbb{L}) := \sum_{P \in \mathbb{P}} \frac{|P|}{|D|} \max_{L \in \mathbb{L}} \text{Precision}(P, L).$$

For some selected parameter combinations that proved to be very good wrt. purity, we also investigated their

$$\text{InversePurity}(\mathbb{P}, \mathbb{L}) := \sum_{L \in \mathbb{L}} \frac{|L|}{|D|} \max_{P \in \mathbb{P}} \text{Precision}(L, P).$$

Both measures have the interval $[0, 1]$ as range. Their difference is that purity measures the purity of the resulting clusters when evaluated against a pre-categorization, while inverse purity measures how stable the pre-defined cate-

[7] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[8] http://www4.fao.org/faobib/index.html

[9] http://www.fao.org/agrovoc/

gories are when split up into clusters. Thus, purity achieves an "optimal" value of 1 when the number of clusters k equals $|D|$, whereas inverse purity achieves an "optimal" value of 1 when k equals 1. Another name in the literature for inverse purity is microaveraged precision. The reader may note that, in the evaluation of clustering results, microaveraged precision is identical to microaveraged recall (cf. e.g. (Sebastiani, 2002)).

4.3. The Reuters-Corpus

We have performed all evaluations on the Reuters-21578 document set. In order to be able to perform comparisons with an *a priori* categorization, we have restricted ourselves to the 12344 documents that were manually classified by Reuters. Documents in the manually classified set were labeled with zero, one, or more of the 135 pre-defined categories.[10] The lack of a label indicates that the human annotator could not find an adequate category. We gathered all the documents without any category label into a new category "defnoclass".[11]

Standard measures like purity (or mutual information or entropy) only allow for the comparison of two partitionings, but they do not allow for the comparison of structures when documents are manually assigned to *several* categorizations and/or documents are automatically assigned to *multiple* clusters. Therefore, we have only selected the first label of each document and ended up with a categorization of the documents into overall 82 categories, including "defnoclass".

To be able to perform evaluations for more different parameter settings, we have restricted the number of documents from the corpus. First, categories with extremely few documents have been discarded with the minimum amount of 15 — thus, "outlier categories" are ignored in the evaluation.[12] Second, we have restricted the category sizes to max. 100 documents by sampling. We call the resulting corpus PRC-min15-max100. It consists of 46 categories and 2619 documents with an average of 56.93 documents per category (standard deviation of 33.12). The text document representation consists of term vectors of length 1219 to 9924 and concept vectors (or mixed term/concept vectors) of length 1468 to 16157, depending on the applied strategy.

5. Results

Each evaluation result described in the following denotes an average from 20 test runs performed on the given corpus for a given combination of parameter values with randomly chosen initial values for Bi-Section-KMeans. The results we report here have been achieved for $k = 60$ clusters. Varying the number k of clusters for the parameter combinations described below has not altered the overall picture.

On the results we report in the text, we have applied t-tests to check for significance with a confidence of 99.5%. All differences that are mentioned below are significant within a confidence of $\alpha = 0.5\%$.

5.1. Clustering without Background Knowledge

Without background knowledge, averaged purity values ranged from 46.1% to 57% (cf. Figure 1). We have observed that tfidf weighting decisively increased purity values irrespective of what the combination of parameter values was (see for instance Figure 1). Pruning with a threshold of 5 or 30 has not always shown an effect. But it increased always purity values when it was combined with tfidf weighting.

5.2. Clustering with Background Knowledge

For clustering using background knowledge, we have also performed pruning and tfidf weighting as described above. The thresholds and modifications have been enacted on concept frequencies (or mixed term/concept frequencies) instead of term frequencies only. We have computed the purity results for varying parameter combinations as described before.

A subset of all cross evaluations is depicted in Figure 1. Each data point indicates a combination of values as follows:

X-axis: On the X-axis, different parameter combinations are indicated. From bottom to top there are:

- Without background knowledge (Section 2) vs. with background knowledge (Section 3), (Ontology = false/true).
- No use of hypernyms (r=0) vs. five levels of hypernyms added to concept frequencies ($r = 5$), cf. Section 3.4 (Hypdepth = 0 / 5).
- Disambiguation strategy: All concepts / First concept / Disambiguation by context; cf. Section 3.3 (Hypdis = All/First/Context).
- Add Concepts vs. Replace Terms by Concepts vs. Concept Vector Only; cf. Section 3.2 (Hypint = add/repl/only).

[10] The categories are called "topics" in Reuters-21578. To be more general, we will refer to them as "category" in the sequel.

[11] The 12344 documents are indicated by an attribute "TOPIC" set to yes and contain the text surrounded by the "BODY" tag.

[12] We investigate in the technical report (Hotho et al., 2003) the influence of the 36 discarded outlier categories with their overall 136 documents. We observe a 2% lower purity for both the best baseline as well as for the results with background knowledge. The general results are the same.

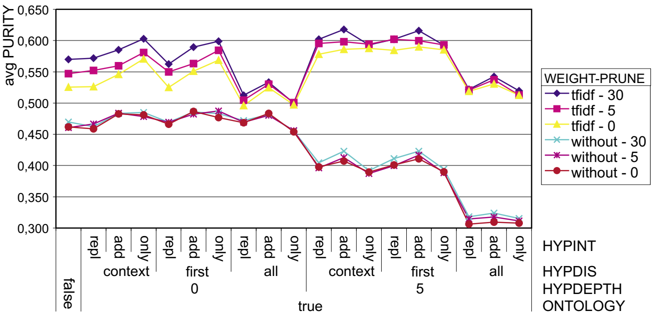

Figure 1. Comparing clustering without background knowledge (leftmost column) against various combinations of parameter settings using background knowledge on PRC-min15-max100 with $k = 60$.

Y-axis: On the Y-axis the resulting purity averaged over 20 test runs for each data point is shown.

Different Lines represent different combinations of tfidf weighting / no weighting with different pruning thresholds (0 vs. 5 vs. 30).

Results. The baseline, i. e., the representation without background knowledge, is given by the best value, 57%, in the leftmost sector (the one for tfidf weighting and a pruning threshold of 30 in Figure 1). The best overall value is achieved by the following combination of strategies: Background knowledge with five levels of hypernyms ($r = 5$), using “disambiguation by context” and term vectors extended by concept frequencies. Purity values then reached 61.8%, thus yielding a relative improvement of 8.4% compared to the baseline.

Without the application of tfidf weighting, all different parameter combinations achieve lower values. Also the difference between the best baseline result (47%) and the best results achieved by adding background knowledge (48,6%) decreases considerably. Furthermore, strategies that consider hypernyms without weighting, like $r = 5$ without tfidf weighting, even decrease the purity compared to the baseline.

Inverse Purity. As may be seen from the description in Section 4.2, purity does not discount evaluation results when splitting up large categories. Therefore, we have investigated how the inverse purity values would be affected for the best baseline (in terms of purity) and a typically good strategy based on background knowledge (again measured in terms of purity). Table 1 summarizes the results favoring background knowledge over the baseline by 51.4% over 47.9%.

ONTO	HYPDEPTH	HYPINT	Purity avg ± std	InversePurity avg ± std
false			**0,57 ± 0,019**	**0,479 ± 0,016**
true	0	add	0,585 ± 0,014	0,492 ± 0,017
		only	0,603 ± 0,019	0,504 ± 0,021
	5	add	**0,618 ± 0,015**	**0,514 ± 0,019**
		only	0,593 ± 0,01	0,500 ± 0,016

Table 1. Results on PRC-min15-max100 $k = 60$, prune=30 (with background knowledge also HYPDIS = context, avg denotes average over 20 cluster runs and std denotes standard deviation)

Inverse Purity and Variance Analysis. We also investigated when and why background knowledge improves the results of Bi-Section-KMeans by analyzing the within-class variance of the Reuters categorization of PRC-min15-max100. For $X \subseteq D$ the variance is defined as:

$$\mathrm{var}(X) := \sum_{d \in X} ||\vec{t_d} - \vec{t_X}||^2 \quad .$$

Based on this, we define the normalized variance within a class L as follows, where the denominator performs a normalization adjusting the variance to the corresponding overall variance of D:

$$\mathrm{var}_{in}(L) := \frac{\mathrm{var}(L)}{\mathrm{var}(D)} \quad .$$

This variance can be computed both for vector representations with and without background knowledge. We thus obtain two values for each class L, namely $\mathrm{var}_{in}^{with}(L)$ and $\mathrm{var}_{in}^{without}(L)$.[13] The normalized difference of the variances is obtained by

$$\mathrm{vd}(L) := \frac{\mathrm{var}_{in}^{with}(L) - \mathrm{var}_{in}^{without}(L)}{\mathrm{var}_{in}^{without}(L)} \quad .$$

The decreasing line in Figure 2 shows this normalized difference of the within-class variance between the representations with (strategy hypdepth=5, hypint=add, hypdis=context, prune=30) and without background knowledge. As becomes evident, for the large majority of pre-defined categories, background knowledge reduces the within-class variance, and hence makes them easier to identify for clustering algorithms which aim at minimizing variance, like Bi-Section-KMeans.

Exceptions can be found when the category is characterized best by syntactic means (e.g., the category “earn” may best be clustered by stop words like ‘vs.’ which are not contained in Wordnet; see leftmost category in Fig 2).

Furthermore, there is a clear tendency that a smaller variance within predefined categories goes along with a higher inverse purity compared to the best baseline. This tendency becomes evident when one compares the variance difference against the individual inverse purity values

$$\mathrm{ipv}(L, \mathbb{P}) := \max_{P \in \mathbb{P}} \mathrm{Precision}(L, P)$$

— which again can be computed with (ipv^{with}) and without ($\mathrm{ipv}^{without}$) background knowledge. This comparison is done in Figure 2 by comparing the variance difference against the inverse purity difference

$$\mathrm{ipd}(L) := \frac{\mathrm{ipv}^{with}(L,\mathbb{P}) - \mathrm{ipv}^{without}(L,\mathbb{P})}{\mathrm{ipv}^{without}(L,\mathbb{P})}$$

[13] Observe that in var_{in} both $\mathrm{var}(L)$ and $\mathrm{var}(D)$ change when background knowledge is incorporated.

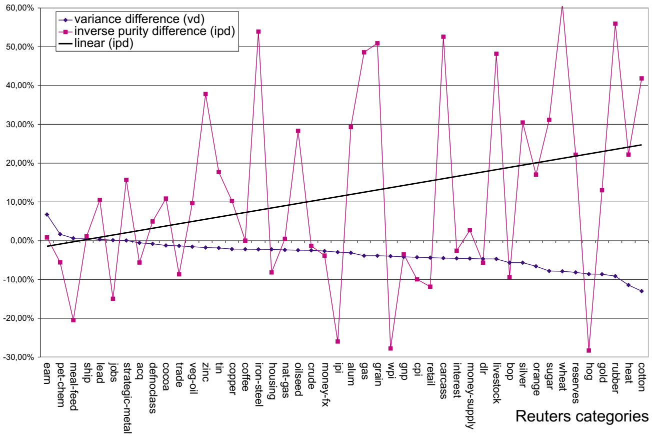

Figure 2. Comparing the variance difference for each given category against the change of clustering results in terms of individual inverse purity values when the preprocessing strategy changes from best baseline to 'standard' (good) background knowledge (strategy hypdepth=5, hypint=add, hypdis=context, prune=30) on PRC-min15-max100 with $k = 60$.

and against its linear interpolation. The diagram shows that the linear interpolation increases with decreasing variance difference. The correlation coefficient of -0.34 between variance difference and individual inverse purity supports this observation.

We have analyzed the categories whose identification by the cluster algorithm is not positively influenced by background knowledge according to the inverse purity difference. Besides of the ones for which within-class variance is not reduced, problems occur for categories that have semantic overlap. For instance, 'dlr', and 'money-fx' are all about money and finance and often co-occur (as second or third Reuters label).

A measure considering also the second and third Reuters label (which is not possible with standard measures like purity) would probably even indicate a positive influence of background knowledge on the clustering.

6. Related Work

While we do not know of any research that exploits background knowledge for text document clustering, there are a number of related uses.

Wordnet has mostly been used in information retrieval and in supervised learning scenarios up to now: In information retrieval, Voorhees (1994) as well as Moldovan and Mihalcea (2000) have explored the possibility to use Wordnet for retrieving documents by keyword search. It has already become clear by their work that particular care must be taken in order to improve precision and recall.

Buenaga Rodríguez et. al. (2000) and Ureña Lóez et. al. (2001) show a successful integration of the Wordnet resource for a document categorization task. They use the Reuters corpus for evaluation and improve the classification results of the Rocchio and Widrow-Hoff algorithms by 20 points. In (Gonzalo et al., 1998), Wordnet is used for word sense disambiguation. They show in an information retrieval setting the improvement of the disambiguated synset model over the term vector model. In contrast to our approach, (de Buenaga Rodrıguez et al., 2000), (Ureña Lóez et al., 2001), and (Gonzalo et al., 1998) apply Wordnet to a supervised scenario (and not to an unsupervised one as in our application), do not make use of Wordnet relations such as hypernyms, and build the term vectors manually.

Approaches like term clustering (Karypis & Han, 2000), LSI (Deerwester et al., 1990) or PLSI (Hofmann, 1999) use statistic methods to compute a kind of "concepts". These concepts are rather different to our definition of ontology concepts. They are not able to indicate the meaning of the concepts and there exists no understandable mapping to lexical entries. A generalization of their 'concepts' is not possible. We do not know of actual comparisons that relate KMeans or Bi-Section-KMeans with LSI or PLSI using the same dataset for clustering.

We have built our numerical comparisons on Bi-Section-KMeans which has proved to be very robust in a wide variety of experiments (Steinbach et al., 2000). Also to our experience it performed as good as other algorithms that we tested informally. Its standard parameter settings evaluated as good as other ones (e. g., bi–secting based on variance instead of cardinality; cf. (Steinbach et al., 2000)).

7. Conclusion

In this paper, we have discussed a way of incorporating background knowledge into a representation for text document clustering in order to improve clustering results. We have performed evaluations on the Reuters data set indicating good performance.

In particular, we found that the best background knowledge strategy (e.g., hypint = add, hypdis = context, hypdepth = 5) can be safely used, as it always improves performance compared to the best baseline.

The principal idea of our approach is that the variance of documents within one category is reduced by representation with background knowledge, thus improving results of text clustering measured in terms of purity and inverse purity with conventional means like Bi-Section-KMeans. To this end, different, but semantically similar terms in two text documents may contribute to a good similarity rating if they are related via Wordnet synsets or hypernyms.

Our experiments have shown that beneficial effects of background knowledge require some care. I.e. we used word sense disambiguation and feature weighting in order to achieve improvements of clustering results. We conjecture that more advanced word sense disambiguation and feature weighting schemes will further improve effectiveness of text clustering.

In our technical report (Hotho et al., 2003), we describe how to make further use of background knowledge for improving explanation capabilities. There we show how to exploit concept representations along a hierarchy, based on Formal Concept Analysis (Ganter & Wille, 1999) in order to derive commonalities and distinctions between different clustering results. For instance, one example result derived there is that several clusters are about 'food' — some about 'coffee' and some about 'cacao'. This result is achieved without 'food' appearing somewhere in the documents, but by taking advantage of the new representation that incorporates background knowledge.

Acknowledgements. We thank our colleagues Alexander Maedche and Viktor Pekar for many fruitful discussions during the work on this paper. This work has been supported by EU in the IST project Bizon and by BMBF in the project PADLR.

References

Agirre, E., & Rigau, G. (1996). Word sense disambiguation using conceptual density. *Proc. of COLING'96*.

Amati, G., Carpineto, C., & Romano, G. (2001). Fub at trec-10 web track: A probabilistic framework for topic relevance term weighting. *The Tenth Text Retrieval Conference (TREC 2001)*. online publication.

Bozsak et al., E. (2002). Kaon - towards a large scale semantic web. *Proceedings of EC-Web* (pp. 304–313). Aix-en-Provence, France: LNCS 2455 Springer.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*, 391–407.

de Buenaga Rodrıguez, M., Hidalgo, J. M. G., & Díaz-Agudo, B. (2000). Using WordNet to complement training information in text categorization. *Recent Advances in Natural Language Processing II*. John Benjamins.

Ganter, B., & Wille, R. (1999). *Formal concept analysis: Mathematical foundations*. Berlin – Heidelberg: Springer.

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarrán, J. (1998). Indexing with WordNet synsets can improve text retrieval. *Proceedings ACL/COLING Workshop on Usage of WordNet for Natural Language Processing*.

Hofmann, T. (1999). Probabilistic latent semantic indexing. *Research and Development in Information Retrieval* (pp. 50–57).

Hotho, A., Staab, S., & Stumme, G. (2003). *Text clustering based on background knowledge* (Technical Report). University of Karlsruhe, Institute AIFB. 36 pages.

Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, *24*, 1–40.

Karypis, G., & Han, E. (2000). Fast supervised dimensionality reduction algorithm with applications to document categorization and retrieval. *Proc. of CIKM-00* (pp. 12–19). ACM Press.

Lewis, D. (1997). Reuters-21578 text categorization test collection.

Miller, G. (1995). WordNet: A lexical database for english. *CACM*, *38*, 39–41.

Moldovan, D. I., & Mihalcea, R. (2000). Using WordNet and lexical operators to improve internet searches. *IEEE Internet Computing*, *4*, 34–43.

Pantel, P., & Lin, D. (2002). Document clustering with committees. *Proc. of SIGIR'02, Tampere, Finland*.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*, 130–137.

Salton, G. (1989). *Automatic text processing: The transformation, analysis and retrieval of information by computer*. Addison-Wesley.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*, 1–47.

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

Ureña Lóez, L. A., de Buenaga Rodríguez, M., & Hidalgo, J. M. G. (2001). Integrating linguistic resources in tc through wsd. *Computers and the Humanities*, *35(2)*, 215–230.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. *Proceedings of ACM-SIGIR. Dublin, Ireland* (pp. 61–69). ACM/Springer.