Ontologies Improve Text Document Clustering

Andreas Hotho, Steffen Staab, Gerd Stumme
{hotho,staab,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 sets of documents 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 core ontologies as background knowledge into the process of clustering text documents. Our experimental evaluations compare clustering techniques based on pre-categorizations of texts from Reuters newsfeeds and on a smaller domain of an eLearning course about Java. In the experiments, improvements of results by background knowledge compared to a baseline without background knowledge can be shown in many interesting combinations.

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 [7].

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 thesauri and ontologies like WordNet. In order to come up with this result we have performed empirical evaluations. This short paper summarizes the main results, while a more in-depth discussion can be found in [4]. In particular, we analyse our novel clustering technique in depth in order to find explanations of when background knowledge may help.

We compare a baseline with different strategies for representing text documents that take background knowledge into account to various extent (Section 2). 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 [8]. For the evaluation (cf. Section 3), we have investigated two text corpora which both come with a set of categorizing labels attached to the documents, *(i)*, the Reuters corpus on newsfeeds, and *(ii)*, a smaller domain of an eLearning course about Java (henceforth called Reuters and Java dataset, respectively). The evaluation results (cf. Section 4) compare the original classification with the partitioning produced by clustering the different representations of the text documents. Briefly, we report also the results we have achieved for the Java corpus, in conjunction with Wordnet on one hand, and with a domain specific ontology on the other hand.

2 Compiling Background Knowledge into the Text Document Representation

Based on the initial text document representation as a bag of words, we have first applied stopword removal. Then we performed stemming, pruning and tfidf weighting in all different combinations. This also holds for the 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.

The background knowledge we have exploited is given through an ontology like Wordnet. Wordnet assigns words of the English language to sets of synonyms called 'synsets'. We consider the synsets as concepts, and use them to extend the bag-of-word model.

2.1 Term vs. Concepts Vector Strategies

Enriching the term vectors with concepts from the core ontology has two benefits. First it resolves synonyms; and

Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)
0-7695-1978-4/03 $ 17.00 © 2003 IEEE

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 (HYPINT) for adding or replacing terms by concepts:

Add Concepts ("add"[1]). 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 $\vec{c_d} := (\text{cf}(d, c_1), \ldots, \text{cf}(d, c_l))$ is the concept vector with $l = |C|$ and $\text{cf}(d, c)$ 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.

2.2 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. [5]), 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:

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

with $\text{tf}(d, T')$ being the sum of the frequencies[2] of all terms $t \in T$ in document d and with $Ref_C(t)$ being the set of all concepts (synsets) assigned to term t in the ontology.

First Concept ("first"). 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 $Ref_C(t)$, i.e. the most common meaning of t. For the other elements of $Ref_C(t)$, frequencies of concepts are not increased by the occurrence of t. Thus the concept frequency is calculated by: $\text{cf}(d, c) := \text{tf}(d, \{t \in T \mid \text{first}(Ref_C(t)) = c\})$ where $\text{first}(Ref_C)$ gives the first concept $c \in Ref_C$ according to the order from Wordnet.

Disambiguation by Context ("context"). The sense of a term t that refers to several different concepts $Ref_C(t) := \{b, c, \ldots\}$ may be disambiguated by a simplified version of [1]'s strategy: Define the semantic vicinity of a concept c to be the set of all its direct sub- and superconcepts $V(c) := \{b \in C | c \prec b \text{ or } b \prec c\}$. Collect all terms that could express a concept from the conceptual vicinity of c by $U(c) := \bigcup_{b \in V(c)} Ref_C^{-1}(b)$. The function $\text{dis}: D \times T \rightarrow C$ with $\text{dis}(d, t) := \text{first}\{c \in Ref_C(t) \mid c \text{ maximizes } \text{tf}(d, U(c))\}$. disambiguates term t based on the context provided by document d. Now $\text{cf}(d, c)$ is defined by $\text{cf}(d, c) := \text{tf}(d, \{t \in T \mid \text{dis}(d, t) = c\})$.

2.3 Strategies for considering the concept hierarchy

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

[1] These abbreviations are used below in Section 4.2

[2] or tfidf's if this weighting is applied

Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)
0-7695-1978-4/03 $ 17.00 © 2003 IEEE

$H(c,r) := \{c' | \exists c_1, \ldots, c_i \in C : c' \prec c_1 \prec \ldots \prec c_i = c,\ 0 \leq i \leq r\}$ 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.

3 Experimental Setting

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 e.g. on the Reuters corpus. In [8] 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.

In the experiments we have varied the different strategies for plain term vector representation and for vector representations containing background knowledge as elaborated above. We have clustered the representations of the corpora using Bi-Section-KMeans and have compared the pre-categorization with our clustering results using standard measures for this task like purity and F-measure.

The Reuters-Corpus. We have performed most of our evaluations on the Reuters-21578 document set ([6][3]). The reason was that it comprises an *a priori* categorization of documents (which we need for evaluating our approach), 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.

To be able to perform evaluations for more different parameter settings, we derived several different subsets of the Reuters corpus. In this short paper, we focus on a corpus which does not include "outlier categories" with less than 15 documents, and restricts all categories to max. 100 documents by sampling. This corpus, called PRC-min15-max100, consists of 46 categories and 2619 documents with an average of 56.93 documents per category. Our extensive evaluation shows, however, that the results did not change significantly when choosing different subsets.

The Java-Corpus. The Java-Corpus is a small dataset containing web pages of an eLearning course about the programming language Java (cf [2]). There are 94 documents distributed among 8 classes with 2013 different word stems and 20394 words overall.

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

In [3], Nicola Henze has described an ontology for the programming language Java. The ontology has been modeled to support an open, adaptive hypermedia system and consists of 521 concepts and twelve non-taxonomic relations. The maximal depth of the taxonomy is 12 with an average of 6.3. We used this domain specific ontology as another source of background knowledge.

4 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 for the Reuters and $k = 10$ clusters for the java corpus. 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 $\alpha = 0.5\%$.

4.1 Clustering without Background Knowledge on Reuters Dataset

Without background knowledge, averaged purity values for PRC-min15-max100 ranged from 46.1 % to 57 %. We have observed that tfidf weighting decisively increased purity values irrespective of what the combination of parameter values was. Pruning with a threshold of 5 or 30 has not always shown an effect. But it always increased purity values when it was combined with tfidf weighting.

4.2 Clustering with Background Knowledge on Reuters Dataset

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.

Results on Reuters-21578 PRC-min15-max100. The baseline, i. e., the representation without background knowledge with tfidf weighting and a pruning threshold of 30 returns an average purity of 57 %. 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"[4] and term vectors extended by concept

[4] The "first" strategy produced results that were not significantly different.

Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)
0-7695-1978-4/03 $ 17.00 © 2003 IEEE

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

Ontology	HYPDEPTH (r)	HYPINT	Purity avg ± std	InversePurity avg ± std	F-Measure avg ± std	Entropy avg ± std
false			**0,57 ± 0,019**	**0,435 ± 0,016**	**0,479 ± 0,016**	**1,329 ± 0,038**
true	0	add	0,585 ± 0,014	0,449 ± 0,018	0,492 ± 0,017	1,260 ± 0,052
		only	0,603 ± 0,019	0,460 ± 0,020	0,504 ± 0,021	1,234 ± 0,038
	5	add	**0,618 ± 0,015**	**0,473 ± 0,019**	**0,514 ± 0,019**	**1,178 ± 0,040**
		only	0,593 ± 0,01	0,459 ± 0,017	0,500 ± 0,016	1,230 ± 0,039

Table 2. Results on Java dataset for $k = 10$ and $prune = 17$ (with background knowledge also HYPDIS = first, HYPDEPTH = 1, avg denotes average over 20 cluster runs and std denotes standard deviation)

Ontology	HYPINT	Purity avg ± std	InversePurity avg ± std	F-Measure avg ± std	Entropy avg ± std
false		**0,61 ± 0,051**	**0,662 ± 0,062**	**0,602 ± 0,047**	**0,845 ± 0,102**
Wordnet	add	0,634 ± 0,070	0,665 ± 0,051	0,626 ± 0,062	0,803 ± 0,125
Java ontology	add	0,651 ± 0,076	**0,685 ± 0,064**	**0,646 ± 0,061**	**0,745 ± 0,122**
Wordnet	only	0,630 ± 0,052	0,635 ± 0,051	0,610 ± 0,051	0,825 ± 0,093
Java ontology	only	**0,669 ± 0,041**	0,646 ± 0,026	0,637 ± 0,036	0,751 ± 0,085

frequencies. Purity values then reached 61.8%, thus yielding a relative improvement of 8.4% compared to the baseline.

Inverse Purity, F-Measure, Entropy on Reuters-21578 PRC-min15-max100. We observed that purity does not discount evaluation results when splitting up large categories. Therefore, we have investigated how inverse purity, F-measure and entropy 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. It shows, e. g., that background knowledge is favored over the baseline by 51.4% over 47.9% wrt. F-measure, and showing similar relations for inverse purity and entropy.

4.3 Results on Java dataset

In order to assure that our observations do not depend on some specific structure of the Reuters dataset, we also performed our experiments on the Java dataset. The major results are shown in Table 2. They indeed back up our observations gained from the Reuters dataset, as the results on the Java dataset with Wordnet on one hand, and the domain specific ontology on the other hand are analogous to the results on the Reuters corpus. Additionally, we could make two more observations: (1) The amount of hypernyms that should be added depends on the size of the thesaurus: The java ontology is too small to derive worth from more than one level of generalization, HYPDEPTH=1 achieves the best values. (2) An ontology tailored to the domain improves the clustering. The purity, for instance, increases by 1.7 points for the 'add' strategy, and by 3.9 points for the 'only' strategy. The other measures improved as well when using the domain specific ontology.

In this short paper, we could only briefly present the most significant results of our extensive evaluation. More details are given in [4].

References

[1] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proc. of COLING'96*, 1996.

[2] M. Gutschke. Kategorisierung von textuellen lernobjekten mit methoden des maschinellen lernens. Studienarbeit, Universität Hannover, Hannover, 2003.

[3] N. Henze. Towards open adaptive hypermedia. In *9. ABIS-Workshop 2001, im Rahmen der Workshopwoche "Lernen - Lehren - Wissen - Adaptivität" (LLWA 01)*, Dortmund, 2001.

[4] A. Hotho, S. Staab, and G. Stumme. Text clustering based on background knowledge. Technical Report 425, University of Karlsruhe, Institute AIFB, 2003. 36 pages.

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

[6] D. Lewis. Reuters-21578 text categorization test collection, 1997.

[7] G. Miller. WordNet: A lexical database for english. *CACM*, 38(11):39–41, 1995.

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

Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)
0-7695-1978-4/03 $ 17.00 © 2003 IEEE