Boosting for Text Classification with Semantic Features

Stephan Bloehdorn[1], Andreas Hotho[2]

[1] University of Karlsruhe, Institute AIFB, Knowledge Management Group, Germany
sbl@aifb.uni-karlsruhe.de
[2] University of Kassel, Knowledge and Data Engineering Group, Germany
hotho@cs.uni-kassel.de

Abstract. Current text classification systems typically use term stems for representing document content. Ontologies allow the usage of features on a higher semantic level than single words for text classification purposes. In this paper we propose such an enhancement of the classical document representation through concepts extracted from background knowledge. Boosting, a successful machine learning technique is used for classification. Comparative experimental evaluations in three different settings support our approach through consistent improvement of the results. An analysis of the results shows that this improvement is due to two separate effects.

1 Introduction

Most of the explicit knowledge assets of today's organizations consist of unstructured textual information in electronic form. Users are facing the challenge of organizing, analyzing and searching the ever growing amounts of documents. Systems that automatically classify text documents into predefined thematic classes and thereby contextualize information offer a promising approach to tackle this complexity. During the last decades, a large number of machine learning methods have been proposed for text classification tasks [16]. Recently, especially Support Vector Machines [9] and Boosting Algorithms [15] have produced promising results.

So far, however, existing text classification systems have typically used the *Bag-of-Words model* known from information retrieval, where single words or word stems are used as features for representing document content. By doing so, the chosen learning algorithms are restricted to detecting patterns in the used *terminology* only, while *conceptual* patterns remain ignored. Specifically, systems using only words as features exhibit a number of inherent deficiencies:

1. *Multi-Word Expressions* with an own meaning like *"European Union"* are chunked into pieces with possibly very different meanings when treated separately like – in this example – *"union"*.
2. *Synonymous Words* like *"tungsten"* and *"wolfram"* are mapped into different features.
3. *Polysemous Words* are treated as one single feature while they may actually have multiple distinct meanings.
4. *Lack of Generalization*: there is no way to generalize similar terms like "gold" and "silver" to their common hypernym "precious metal".

While items 1 – 3 directly address issues that arise on the lexical level, item 4 rather addresses an issue that is situated on a conceptual level. In this paper, we show how background knowledge in form of simple ontologies can improve text classification results by directly addressing these problems.

We propose a hybrid approach for document representation based on the common term stem representation which is enhanced with concepts extracted from the used ontologies. For actual classification we suggest to use the AdaBoost algorithm which has proven to produce accurate classification results in many experimental evaluations and seems to be well suited to integrate different types of features. Evaluation experiments on three text corpora, namely the Reuters-21578, OHSOMED and FAODOC collections show that our approach leads to consistent improvements of the results. We also show that in most cases the improvement can be traced to two distinct effects, one being situated mainly on the lexical level and the generalization on the conceptual level.

This paper is organized as follows. We introduce some preliminaries, namely the classical bag-of-words document representation and ontologies in section 2. A detailed process for compiling conceptual features into an enhanced document representation is presented in section 3. In section 4 we review the AdaBoost algorithm and its inner workings. Evaluation Measures for text classification are reviewed in section 5. In the following, experimental evaluation results of our approach are presented for the Reuters-21578, OHSOMED, and FAODOC corpora under varying parameter combinations. It turns out that combined feature representations perform consistently better than the pure term-based approach. We review related work in section 7 and conclude with a summary and outlook in section 8.

2 Preliminaries

The Bag-Of-Words Paradigm In the common term-based representation, documents are considered to be bags of terms, each term being an independent feature of its own. Let D be the set of documents and $T = \{t_1, \ldots, t_m\}$ the set of all different terms occurring in D. For each term $t \in T$ in document $d \in D$ one can define feature values functions like binary indicator variables, absolute frequencies or more elaborated measures like TFIDF [14].

Typically, whole words are not used as features. Instead, documents are first processed with stemming algorithms, e.g. the Porter stemmer for English [13]. In addition, *Stopwords*, i.e. words which are considered as non–descriptive within a bag–of–words approach, are typically removed.

Ontologies The background knowledge we have exploited is given through simple ontologies. We first describe the structure of these ontologies and then discuss their usage for the extraction of conceptual feature representations for text documents. The background knowledge we will exploit further on is encoded in a *core ontology*. For the purpose of this paper, we present only those parts of our more extensive ontology definition [2] that we need within this paper.

Definition 1 (Core Ontology). *A core ontology is a structure* $\mathcal{O} := (C, <_C)$ *consisting of a set C, whose elements are called concept identifiers, and a partial order* $<_C$ *on C, called concept hierarchy or taxonomy.*

Definition 2 (Subconcepts and Superconcepts). *If* $c_1 <_C c_2$ *for any* $c_1, c_2 \in C$*, then* c_1 *is a subconcept (specialization) of* c_2 *and* c_2 *is a superconcept (generalization) of* c_1*. If* $c_1 <_C c_2$ *and there exists 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*, denoted by* $c_1 \prec c_2$*.*

These specialization/generalization relationships correspond to what we know as is-a vs. is-a-special-kind-of, resulting in a hierarchical arrangement of concepts[3]. In ontologies that are more loosely defined, the hierarchy may, however, not be as explicit as is-a relationships but rather correspond to the notion of narrower-than vs. broader-than[4]

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 3 (Lexicon for an Ontology). *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 (symbols), 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*, for* $s \in S_C$ *we define* $Ref_C(s) := \{c \in C | (s, c) \in Ref_C\}$*. Analogously, for* $c \in C$ *it is* $Ref_C^{-1}(c) := \{s \in S_C | (s, c) \in Ref_C\}$*. An ontology with lexicon is a pair* $(\mathcal{O}, Lex)$ *where* $\mathcal{O}$ *is an ontology and* Lex *is a lexicon for* $\mathcal{O}$*.*

Ontologies for the experimental evaluation For the purpose of actual evaluation in the experiments, we have used three different resources, namely WordNet, the MeSH Tree Structures Ontology, and the AGROVOC ontology.

Although not explicitly designed as an ontology, *WordNet* [12] largely fits into the ontology definitions given above. The WordNet database organizes simple words and multi-word expressions of different syntactic categories into so called *synonym sets (synsets)*, each of which represents an underlying concept and links these through semantic relations. The current version 2.0 of WordNet comprises a total of 115,424 synsets and 144,309 lexical index terms. The noun category, which was the main focus of our attention[5], contains nearly 70 % of the total synsets, links from 114,648 index terms to 79,689 synsets in a total of 141,690 mappings. The collection of index terms in WordNet comprises base forms of terms and their exceptional derivations. The retrieval of base forms for inflected forms is guided by a set of category-specific morphological

[3] Note that this hierarchical structure is not necessarily a tree structure. It may also be a *directed acyclic graph* possibly linking concepts to multiple superconcepts at the same time.

[4] In many settings this view is considered as a very bad practice as it may lead to inconsistencies when reasoning with ontologies. However, this problem does not arise in the context of this work.

[5] Beside the noun category, we have also exploited verb synsets, however, without making use of any semantic links,

transformations, which ensure a high precision in the mapping of word forms to index words.

The MeSH Tree Structures Ontology is an ontology that has been compiled out of the Medical Subject Headings (MeSH) controlled vocabulary thesaurus of the United States National Library of Medicine (NLM). The ontology contains more than 22,000 concepts, each enriched with synonymous and quasi-synonymous language expressions. The underlying hierarchical structure is in large parts consistent with real hypernym relations but also comprises other forms of hierarchical arrangements. The ontology itself was ported into and accessed through the Karlsruhe Ontology and Semantic Web Infrastructure (KAON) infrastructure[6].

The third ontology that has been used is the AGROVOC Ontology, based on AGROVOC, a multilingual agricultural thesaurus[7] developed by the United Nations Food and Agricultural Organization (FAO). In total, the ontology comprises 17,506 concepts from the agricultural domain. The lexicon contains label and synonym entries for each concept in English and six additional languages. The concept hierarchy in the AGROVOC ontology is based on broader-term relationships thus not necessarily on strict superconcept relations in some cases.

3 Conceptual Document Representation

To extract concepts from texts, we have developed a detailed process, that can be used with any ontology with lexicon. The overall process comprises five processing steps that are described in this section.

Candidate Term Detection Due to the existence of multi-word expressions, the mapping of terms to concepts cannot be accomplished by querying the lexicon directly for the single words in the document.

We have addressed this issue by defining a candidate term detection strategy that builds on the basic assumption that finding the longest multi-word expressions that appear in the text and the lexicon will lead to a mapping to the most specific concepts. The candidate expression detection algorithm we have applied for this lookup procedure is given in algorithm 1[8].

The algorithm works by moving a window over the input text, analyze the window content and either decrease the window size if unsuccessful or move the window further. For English, a window size of 4 is sufficient to detect virtually all multi-word expressions.

Syntactical Patterns Querying the lexicon directly for any expression in the window will result in many unnecessary searches and thereby in high computational requirements. Luckily, unnecessary search queries can be identified and avoided through an analysis of the part-of-speech (POS) tags of the words contained in the current window. Concepts are typically symbolized in texts within *noun phrases*. By defining appropriate

[6] see `http://kaon.semanticweb.org/`

[7] see `http://www.fao.org/agrovoc/`

[8] The algorithm here is an improved version of one proposed in [17].

Algorithm 1 The candidate expression detection algorithm

```
Input: document $d = \{w_1, w_2, \ldots, w_n\}$,
  $Lex = (S_C, Ref_C)$ and window size $k \geq 1$.
  $i \leftarrow 1$
  list $L_s$
  index-term s
  while $i \leq n$ do
    for $j = min(k, n - i + 1)$ to 1 do
      $s \leftarrow \{w_i \ldots w_{i+j-1}\})$
      if $s \in S_C$ then
        save s in $L_s$
        $i \leftarrow i + j$
        break
      else if $j = 1$ then
        $i \leftarrow i + j$
      end if
    end for
  end while
  return $L_s$
```

POS patterns and matching the window content against these, multi-word combinations that will surely not symbolize concepts can be excluded in the first hand and different syntactic categories can be disambiguated.

Morphological Transformations Typically the lexicon will not contain all inflected forms of its entries. If the lexicon interface or separate software modules are capable of performing base form reduction on the submitted query string, queries can be processed directly. For example, this is the case with WordNet. If the lexicon, as in most cases, does not contain such functionalities, a simple fallback strategy can be applied. Here, a separate index of stemmed forms is maintained. If a first query for the inflected forms on the original lexicon turned out unsuccessful, a second query for the stemmed expression is performed.

Word Sense Disambiguation Having detected a lexical entry for an expression, this does not necessarily imply a one-to-one mapping to a concept in the ontology. Although multi-word-expression support and pos pattern matching reduce ambiguity, there may arise the need to disambiguate an expression versus multiple possible concepts. The *word sense disambiguation (WSD)* task is a problem in its own right [8] and was not the focus of our work.

In our experiments, we have used three simple strategies proposed in [7] to process polysemous terms:

- The ‘‘all” strategy leaves actual disambiguation aside and uses all possible concepts.
- The ‘‘first” strategy exploits WordNet’s capability to return synsets ordered with respect to usage frequency. This strategy chooses the most frequent concept in case of ambiguities.
- The ‘‘context” strategy performs disambiguation based on the degree of overlap of lexical entries for the semantic vicinity of candidate concepts and the document content as proposed in [7].

Generalization The last step in the process is about going from the specific concepts found in the text to more general concept representations. Its principal idea is that if a term like ‘arrythmia’ appears, one does not only represent the document by the concept corresponding to ‘arrythmia’, but also by the concepts corresponding to ‘heart disease’ and ‘cardiovascular disease’ etc. up to a certain level of generality. This is realized by compiling, for every concept, all superconcept up to a maximal distance h into the concept representation. Note that the parameter h needs to be chosen carefully as climbing up the taxonomy too far is likely to obfuscating the concept representation.

4 Boosting

Boosting is a relatively young, yet extremely powerful machine learning technique. The main idea behind boosting algorithms is to combine multiple *weak learners* – classification algorithms that perform only slightly better than random guessing – into a powerful composite classifier.

Although being refined subsequently, the main idea of all boosting algorithms can be traced to the first practical boosting algorithm, AdaBoost [4], which we will concentrate on in this paper. AdaBoost and related algorithms have proved to produce extremely competitive results in many settings, most notably for text classification [15]. At the beginning, the inner workings of boosting algorithms were not well understood. Subsequent research in boosting algorithms made them rest on a well developed theoretical framework and has recently provided interesting links to other successful learning algorithms, most notably to Support Vector Machines, and to linear optimization techniques [11].

AdaBoost The idea behind “boosting” weak learners stems from the observation that it is usually much easier to build many simple “rules of thumb” than a single highly complex decision rule. Very precise overall decisions can be achieved if these weak learners are appropriately combined.

This idea is reflected in the output of the boosting procedure: for AdaBoost the aggregate decisions are formed in an *additive model* of the form:

$$\hat{f}(x) = \mathrm{sign}(\sum_{t=1}^{T} \alpha_t \, h_t(x))$$

with $h_t : \mathbb{X} \to \{-1, 1\}, \;\; \alpha_t \in \mathbb{R}$, where α_t denotes the weight of the ensemble member h_t in the aggregate decision and where the output values $\hat{f}(x) \in \{1, -1\}$ denote positive and negative predictions respectively. In such a model, AdaBoost has to solve two questions: How should the set of base hypotheses h_t be determined ? How should the weights α_t determined, i.e. which base hypotheses should contribute more than others and how much ? The AdaBoost algorithm, described in algorithm 2 aims at coming up with an optimal parameter assignment for h_t and α_t.

AdaBoost maintains a set of weights D_t over the training instances $x_1 \ldots x_i \ldots x_n$. At each iteration step t, a base classifier is chosen that performs best on the *weighted* training instances. Based on the performance of this base classifier, the final weight

Algorithm 2 The AdaBoost algorithm.

```
Input: training sample $\mathcal{S}_{train} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
  with $(x_i, y_i) \in \mathbb{X} \times \{-1, 1\}$ and $y_i = f(x_i)$,
  number of iterations $T$.
Initialize: $D_1(i) = \frac{1}{n}$ for all $i = 1, \ldots, n$.
  for $t = 1$ to $T$ do
    train base classifier $h_t$ on weighted training set
    calculate the weighted training error:
        $\epsilon_t \leftarrow \sum_{i=1}^{n} D_t(i)\, I_{y_i \neq h_t(x_i)}$    (1)
    compute the optimal update step as:
        $\alpha_t \leftarrow \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$    (2)
    update the distribution as:
        $D_{t+1}(i) \leftarrow \frac{D_t(i)\, e^{-\alpha_t\, y_i\, h_t(x_i)}}{Z_t}$    (3)
    where $Z_t$ is a normalization factor to ensure that $\sum_{i=1}^{n} D_{t+1}(i) = 1$
    if $\epsilon_t = 0$ or $\epsilon_t = \frac{1}{2}$ then
      break
    end if
  end for
  Result: composite classifier given by:
        $\hat{f}(x) = \mathrm{sign}\left(\hat{f}_{soft}(x)\right) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$    (4)
```

parameter α_t is calculated in equation (2) and the distribution weights D_{t+1} for the next iteration are updated. The weight update in equation (3) assigns higher weights to training instances that have been misclassified, while correctly classified instances will receive smaller weights in the next iteration. Thereby, AdaBoost kind of "focusing in" on those examples that are more difficult while the weight each base classifier receives in the final additive model depends on its performance on the weighted training set at the respective iteration step.

Weak Lerners for AdaBoost In theory, AdaBoost can be used with *any* base learner capable of handling weighted training instances. Although the base classifiers are not restricted to belong to a certain classifier family, virtually all work with boosting algorithms has used the very simple class of *decision stumps* as base learners.

In this presentation, we focus on simple indicator function decision stumps of the form:

$$h(x) = \begin{cases} c \text{ if } x^j = 1 \\ -c \text{ else.} \end{cases}$$

with $c \in \{-1, 1\}$. A decision stump of this form takes binary features (e.g. word or concept occurrences) as inputs. The index j identifies a specific binary feature whose presence either supports a positive classification decision, i.e. $c = 1$ or a negative decision, i.e. $c = -1$.

5 Evaluation Metrics

A standard set of performance metrics is commonly used to assess classifier performance which we will review shortly in this section.

Classification Metrics Given a set of test documents $\mathcal{S} = \{x_1, \ldots, x_n\}$ with binary labels $\{y_1, \ldots, y_n\}$ where $y_i \in \{-1, 1\}$ codes the membership in a class in question. Given further a classifier $\hat{f}$ trained on an independent training set with $\hat{f}(x) \in \{-1, 1\}$ indicating the binary decisions of the classifier. Then the test sample can be partitioned into sets $\mathcal{S} = \mathcal{S}^+ \cup \mathcal{S}^-$, i.e. the set of positive and negative test documents. These partitions can be decomposed further into $\mathcal{S}^+ = TP \cup FN$ and $\mathcal{S}^- = FP \cup TN$ with: $TP = \{x_i \in \mathcal{S} | \hat{f}(x_i) = 1 \wedge y_i = 1\}$, $FP := \{x_i \in \mathcal{S} | \hat{f}(x_i) = 1 \wedge y_i = -1\}$, $TN := \{x_i \in \mathcal{S} | \hat{f}(x_i) = -1 \wedge y_i = -1\}$ and $FN := \{x_i \in \mathcal{S} | \hat{f}(x_i) = -1 \wedge y_i = 1\}$ called the sets of documents classified *true positive*, *false positive*, *true negative* and *false negative*, often referred to as the classification contingency table.

Based on these definitions, different evaluation measures have been defined [18]. Commonly used classification measures in text classification and information retrieval are the *classification error*, *precision*, *recall* and the F_β *measure*:

1. Classification Error

$$err(\hat{f}, \mathcal{S}) := \frac{|FP| + |FN|}{|TP| + |FP| + |TN| + |FN|} \ . \tag{5}$$

2. Precision

$$prec(\hat{f}, \mathcal{S}) := \frac{|TP|}{|TP| + |FP|} \ . \tag{6}$$

3. Recall

$$rec(\hat{f}, \mathcal{S}) := \frac{|TP|}{|TP| + |FN|} \ . \tag{7}$$

4. F_1 measure

$$F_1(\hat{f}, \mathcal{S}) := \frac{2 \, prec(\hat{f}, \mathcal{S}) \; rec(\hat{f}, \mathcal{S})}{prec(\hat{f}, \mathcal{S}) + rec(\hat{f}, \mathcal{S})} \ . \tag{8}$$

Ranking Metrics The ensemble classifiers produced by AdaBoost are capable of returning a real-valued output $\hat{f}_{soft}(x) \in [-1, 1]$. The magnitude $|\hat{f}_{soft}(x)|$ reflects the "confidence" of the classifier in a decision and allows to rank documents. Consequently, a parameterized classifier $\hat{f}_k$ can be defined that returns $\hat{f}_k(x) = 1$ if $\hat{f}_{soft}(x)$ ranks among the first k documents and $\hat{f}_k(x) = -1$ otherwise. On this basis, values for precision and recall can be calculated and tuned with respect to different values of k. When precision and recall coincide at some k, this value is called the break-even point (BEP). It can be shown that this is necessarily the case at $k = |\mathcal{S}^+|$[9].

Micro- and Macro Averaging To average evaluation results over binary classifications on the per-class level, two conventional methods exist. The *macro-averaged* figures are meant to be averages on the class level and are calculated as simple averages of the scores achieved for the different classes. In contrast, *micro-averaged* figures are computed by summing the cells of per-class contingency tables together and then computing performance scores based on these global figures. These can consequently be seen as averages on the document level.

Statistical Significance Tests Statistical significance tests are useful in order to verify to which extent the claim of an improvement can be backed by the observations on the test set. For the experiments we report in this paper, we focused on two statistical significance tests, a sign test ("S-test") and a paired t-test ("T-test") on an improvement of individual F_1 scores for the different classes that have been evaluated in each experiment described in detail in [19]. Following common statistical practice, we have required a significance level $\alpha = 0.05$ is required for claiming an improvement to be *significant*. The significance level of $\alpha = 0.01$ was used for the claim that an improvement was *very significant*.

6 Experiments

The focus of our evaluation experiments was directed towards comparing whether AdaBoost using the enhanced document representation would outperform the classical term representation.

6.1 Evaluation on the Reuters-21578 Corpus

A first set of evaluation experiments was conducted on the well-known Reuters-21578 collection. We used the "ModApte" split which divides the collection into 9,603 training documents, 3,299 test documents and 8,676 unused documents.

[9] This follows from the fact that if there are m negative documents among the first $|\mathcal{S}^+|$ documents in the ranked list, there must also be exactly m positive examples in the remainder of the list, thus: $FP_k = FN_k = m$, which guarantees precision and recall to be equal according to the formulas given above.

Experimental Setup In the first stage of the experiment, terms and concepts were extracted as features from the documents in the training and test corpus. For terms, the feature extraction stage consisted of the stages described in section 2, namely chunking, removal of the standard stopwords for English defined in the stopword list from the SMART system containing 571 words [10] and stemming using the porter stemming algorithm, resulting in a total number of 17,525 distinct term features. Conceptual features were then extracted for noun and verb phrases using WordNet as background ontology. Different sets of concept features were extracted based on varying parameters for disambiguation strategy and maximal hypernym distance ranging from 10,259 to 27,236 distinct concept features.

In the next stage of the experiment, classification was performed using AdaBoost. We performed binary classification on the top 50 categories containing the highest number of positive training documents. The number of boosting iterations for training was fixed at 200 rounds for all feature combinations.

Results As a general finding, the results obtained in the experiments suggest that AdaBoost typically achieves better classification for both macro- and micro-averaged results when used with a combination of term-based and concept-based features. Table 1 summarizes the results of the experiments for different feature types with the best values being highlighted. The relative gains on the F_1 value, which is influenced both by precision and recall, compared to the baseline show that in all but one cases the performance can be improved by including conceptual features, peaking at an relative improvement of 3.29 % for macro-averaged values and 2.00 % for micro-averaged values. Moderate improvements are achieved through simple concept integration, while larger improvements are achieved in most cases through additional integration of more general concepts.

The results of the significance tests allow us to conclude that these improvements are significant in at least half of the cases. In general, the improvements of macro-averaged F_1 are higher than with micro-averaging which seems to suggest that the additional concepts are particularly helpful for smaller classes.

6.2 Evaluation on the OHSOMED Corpus

A second series of experiments was conducted using the OHSOMED collection, initially compiled by Hersh et al. [6]. It consists of titles and abstracts from medical journals, each being indexed with multiple MeSH descriptors. We have used the 1987 portion of the collection containing a total of 54,708 entries. Two thirds of the entries were randomly selected as training documents while the remainder was used as test set, resulting in a training corpus containing 36,369 documents and a test corpus containing 18,341 documents.

Experimental Setup Term stems were extracted as with Reuters-21578 resulting in a total number of 38,047 distinct features. WordNet and the MeSH Tree Structures Ontology were used to extract conceptual features. For WordNet, noun and verb phrases

[10] see `ftp://ftp.cs.cornell.edu/pub/smart/english.stop`

		macro-averaged (in percentages)			
Feature Type	**Error**	**Prec**	**Rec**	**F$_1$**	**BEP**
term	00.65	80.59	66.30	72.75	74.29
term & synset.first	00.64	80.66	67.39	73.43	75.08
term & synset.first.hyp5	00.60	80.67	**69.57**	74.71	74.84
term & synset.first.hyp10	00.62	80.43	68.40	73.93	**75.58**
term & synset.context	00.63	79.96	68.51	73.79	74.46
term & synset.context.hyp5	00.62	79.48	68.34	73.49	74.71
term & synset.all	00.64	80.02	66.44	72.60	73.62
term & synset.all.hyp5	**00.59**	**83.76**	68.12	**75.14**	75.55

		micro-averaged (in percentages)			
Feature Type	**Error**	**Prec**	**Rec**	**F$_1$**	**BEP**
term	00.65	89.12	79.82	84.21	85.77
term & synset.first	00.64	88.75	80.79	84.58	85.97
term & synset.first.hyp5	00.60	89.16	**82.46**	85.68	85.91
term & synset.first.hyp10	00.62	88.78	81.74	85.11	86.14
term & synset.context	00.63	88.86	81.46	85.00	85.91
term & synset.context.hyp5	00.62	89.09	81.40	85.07	85.97
term & synset.all	00.64	88.82	80.99	84.72	85.69
term & synset.all.hyp5	**00.59**	**89.92**	82.21	**85.89**	**86.44**

Table 1. Evaluation Results for Reuters-21578.

were considered while for the MeSH Tree Structures Ontology, only noun phrases were considered.

For WordNet, the same disambiguation strategies were used as in the Reuters-21578 experiments. For the MeSH Tree Structures Ontology, only the “all” strategy was used due to the observation that polysemy problems occur extremely rarely with this ontology as descriptor terms are most naturally unique. For both ontologies, different degrees of depth were used for hypernym or superconcept integration, resulting in a total of 16,442 to 34,529 synset features and 11,572 to 13,663 MeSH concept features.

On the documents of the OHSOMED dataset — as on Reuters-21578 — binary classification with AdaBoost was performed on the top 50 categories that contained the highest number of positive training documents. To cope with the on average larger number of features and the much higher number of documents compared to the Reuters-21578 corpus, the number of boosting iterations for all experiments with the OHSOMED collection was set to 1000 rounds.

Results Different runs of the classification stage were performed based on the different features, leading to often substantially different results. Again, the general finding is that complementing the term stem representation with conceptual features significantly improves classification performance.

Table 2 summarizes the macro- and micro-averaged results. The relative improvements for the F_1 scores compared to the term stem baseline are depicted in figure 6.2 for WordNet as background knowledge resource. These range from about 2% to a maxi-

		macro-averaged (in percentages)			
Feature Type	**Error**	**Prec**	**Rec**	**F$_1$**	**BEP**
term	00.53	52.60	35.74	42.56	45.68
term & synset.first	00.52	53.08	36.98	43.59	46.46
term & synset.first.hyp5	00.52	53.82	38.66	45.00	48.01
term & synset.context	00.52	52.83	37.09	43.58	46.88
term & synset.context.hyp5	**00.51**	**54.55**	**39.06**	**45.53**	**48.10**
term & synset.all	00.52	52.89	37.09	43.60	46.82
term & synset.all.hyp5	00.52	53.33	38.24	44.42	46.73
term & mesh	00.52	53.65	37.56	44.19	47.31
term & mesh.sc1	00.52	52.91	37.59	43.95	46.93
term & mesh.sc3	00.52	52.77	38.06	44.22	46.90
term & mesh.sc5	00.52	52.72	37.57	43.87	47.16

		micro-averaged (in percentages)			
Feature Type	**Error**	**Prec**	**Rec**	**F$_1$**	**BEP**
term	00.53	55.77	36.25	43.94	46.17
term & synset.first	00.52	56.07	37.30	44.80	47.01
term & synset.first.hyp5	00.52	56.84	38.76	46.09	48.31
term & synset.context	00.52	56.30	37.46	44.99	47.34
term & synset.context.hyp5	**00.51**	**58.10**	**39.18**	**46.81**	**48.45**
term & synset.all	00.52	56.19	37.44	44.94	47.32
term & synset.all.hyp5	00.52	56.29	38.24	45.54	46.73
term & mesh	00.52	56.81	37.84	45.43	47.78
term & mesh.sc1	00.52	56.00	37.90	45.20	47.49
term & mesh.sc3	00.52	55.87	38.26	45.42	47.45
term & mesh.sc5	00.52	55.94	37.94	45.21	47.63

Table 2. Evaluation Results for OHSOMED.

mum of about 7 %. The relative F_1 improvements when using the MeSH Tree Structure Ontology, were on the 3% to 5% level in all cases.

The statistical significance tests revealed that in virtually all cases, these improvements can be claimed to be significant and actually even very significant in most cases.

Again, the integration of conceptual features improved text classification results. The relative improvements achieved on OHSOMED are generally higher than those achieved on the Reuters-21578 corpus. This makes intuitively sense as the documents in the OHSOMED corpus are taken from the medical domain. Documents from this domain typically suffer heavily from the problems described in section 2, especially synonymous terms and multi-word expressions. But this is only a first effect. The even better results achieved through hypernym integration with WordNet indicate that also the highly specialized language is a problem that can be remedied through integration of more general concepts.

A comparison between WordNet and the MeSH Descriptor Ontology is hard. On the one hand, without generalization, the domain specific MeSH Tree Structures Ontology is able to achieve slightly better results. Taking into account that the extraction was here bases solely on noun phrases and that WordNet's coverage is much broader,

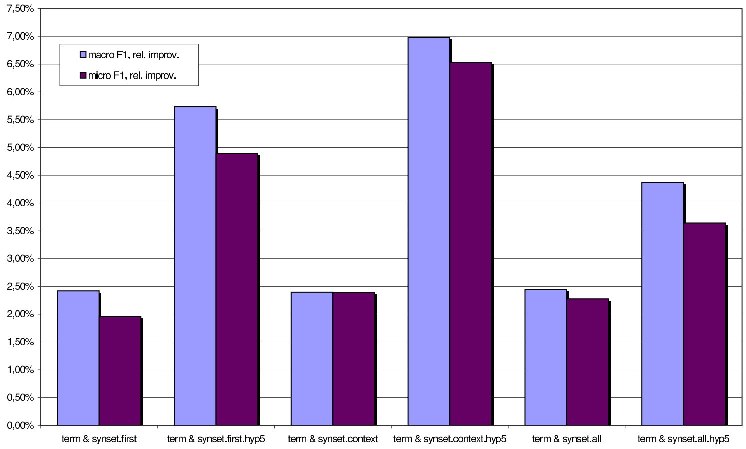

Fig. 1. Relative Improvements of F_1 Scores on OHSOMED for combined Term-Synset Features vs. Term Stems.

this is a positive surprise. On the other hand, WordNet achieves much better results when generalization comes into play. In contrast to WordNet, superconcept integration for MeSH does not really improve the results and varying levels of superconcept integration lead to similar or even worse results. Apparently, the broader-term relation of the MeSH thesaurus is indeed not well suited to improve the results. Also note that in contrast to the Reuters-21578 experiments, "context" word sense disambiguation strategy performs best in combination with hypernym integration. Apparently, it is easier to disambiguate polysemous words in the medical context.

6.3 Evaluation on the FAODOC Corpus

The third and last series of experiments uses a collection of documents from the FAO Document Online Catalogue (FAODOC)[11], managed by the United Nations Food and Agricultural Organization. The FAODOC database houses articles and other publications from the agricultural domain together with metadata information, including subject and category elements.

Experimental Setup The FAODOC collection contains English, French and Spanish HTML documents. All documents are indexed with one or multiple category codes, each of which refers to one of 115 FAODOC subject categories. In the experiments, only the subset of 1,501 English documents has been used where each of the categories has at least 50 positive documents, resulting in 21 distinct subject categories. From the total number of 1,501 documents, the first 1,000 documents were used for training while the remainder of 501 documents were held out as test set. The FAODOC dataset is very different from the other datasets encountered so far. Besides being taken from

[11] see `http://www4.fao.org/faobib/index.html`

		macro-averaged			
Feature Type	**Error**	**Prec**	**Rec**	**F$_1$**	**BEP**
term	06.87	45.47	27.11	33.97	36.93
term & agrovoc	**06.66**	**50.96**	28.63	36.66	39.84
term & agrovoc.sc1	06.76	49.26	27.48	35.28	39.40
term & agrovoc.sc3	06.79	49.08	**30.41**	**37.55**	**41.69**

		micro-averaged			
Feature Type	**Error**	**Prec**	**Rec**	**F$_1$**	**BEP**
term	06.87	50.44	31.22	38.57	44.29
term & agrovoc	**06.66**	**52.91**	32.46	**40.24**	**48.01**
term & agrovoc.sc1	06.76	51.75	**32.60**	40.00	46.77
term & agrovoc.sc3	06.79	51.47	31.36	38.97	47.73

Table 3. Results on FaoDoc

a different domain, the total number of documents is much smaller. The documents in the FaoDoc dataset are typically much larger in size, ranging from 1.5 kilobytes to over 600 kilobytes, which is also reflected in the resulting feature representations with 68,608 word stems. Besides the extraction of term stems as usual, conceptual features were extracted again, this time using the AgroVoc ontology as background knowledge resource. For both types of features, the documents were first converted from HTML to plain text, then proceeding in the same way as with the documents in the other corpora.

As in the other experiments, each of the 21 different labels resulted in a binary classification run of its own, each time using AdaBoost as learning algorithm with decision stump classifier based on the binary feature weights as base learners. The chosen number of 500 boosting iterations is based on a trade-off between the smaller number of training documents on the one hand and a typically larger size per document on the other. In all experiments, the results on the 21 individual labels were eventually macro- and micro-averaged.

Results Different runs of the classification stage were performed based on different features: term stems and again combinations of both types of features.

Table 3 summarizes the results of the experiments with the FaoDoc for the different feature representations, evaluation metrics and averaging variants. For each performance metric, the best result is highlighted. Again, combinations of terms and concepts as features also achieve considerable improvements over the classic term stem representation in all scores, most notably with respect to precision.

Figure 2 undermines the good performance of the term and ‘agrovoc’ concept representation achieving an impressive relative improvement of 10.54 % on the macro-averaged F_1 value compared to the ‘term’ representation. The relative improvement on the micro-averaged F_1 lies at 4.33 %.

As with OhsoMed, one observes a heavy discrepancy between the macro- and micro-averaged scores. Again, macro-averaged performance gains are higher than those for micro-averaging, which makes sense taking into account the fairly unequal category sizes. In contrast to the other experiments, the amount of deviation however varies con-

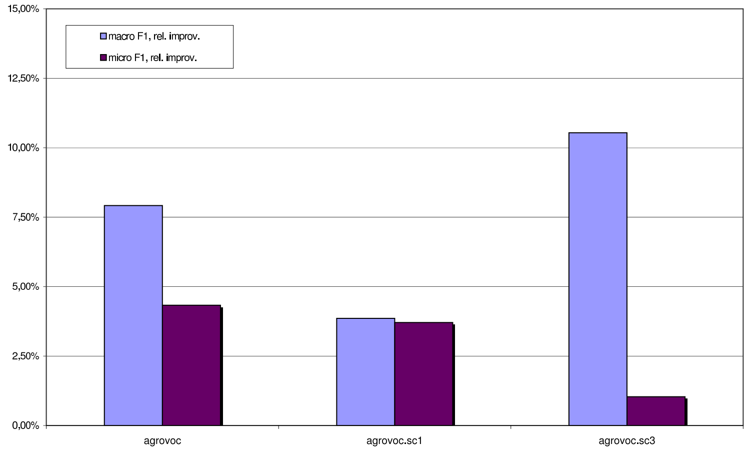

Fig. 2. Bar Chart Illustration of the Relative Improvements of F_1 Scores on all 21 FaoDoc Categories for combined Term-Concept Representations vs. 'term'. All numbers are percentages.

siderably among the different feature representations. Furthermore, the question which superconcept integration depth leads to the best improvement cannot be answered easily because the effects vary between micro- and macro-averaging. We attribute the strong variation in the results to the fact that random effects are much likelier compared to the other experiments as the number of training and test documents was considerably smaller.

7 Related Work

Representing document content through metadata descriptions is a well-known task in the semantic web context, also known as annotation[5]. Typically, however, this is a semi-automatic task that aims at precise metadata descriptions and not at creating features for machine learning algorithms.

To date, the work on integrating semantic background knowledge into text classification or other related tasks is quite scattered. Much of the early work with semantic background knowledge in information retrieval was done in the context of *query expansion* techniques [1]. Feature representations based on concepts from ontological background knowledge were also used in text clustering settings [7] where it could be shown that conceptual representations can significantly improve text cluster purity and reduce the variance among the representations of related documents.

Recent experiments with conceptual feature representations for text classification are presented in [17]. These and other similar published results are, however, still too few to allow insights on whether positive effects can be achieved in general. In some cases, even negative results were reported. For example, a comprehensive comparison of approaches based on different word-sense document representations and different learning algorithms reported in [10] ends with the conclusion of the authors that *"the use of word senses does not result in any significant categorization improvement"*.

Alternative approaches for conceptual representations of text documents that are not based on background knowledge compute kind of "concepts" statistically. Very good results with a probabilistic variant of LSA known as Probabilistic Latent Semantic Analysis (pLSA) were recently reported in [3]. The experiments reported therein are of particular interest as the classification was also based on boosting combined term-concept representation, the latter being however automatically extracted from the document corpus using pLSA.

8 Conclusions

In this paper, we have proposed an approach to incorporate concepts from background knowledge into document representations for text document classification. A very successful ensemble learning algorithm, AdaBoost, was proposed to perform the final classifications based on the classical word vector representations and the conceptual features. Boosting Algorithms, when used with binary feature representations, scale well to a large number of dimensions that typically occur when superconcepts are used as well. At the same time, AdaBoost is capable of integrating heterogenous features that are based on different paradigms without having to adjust any parameters in the feature space representation.

Experiments on three different datasets clearly showed that the integration of concepts into the feature representation clearly improves classification results. The absolute scores achieved on Reuters and OHSOMED are highly competitive with other published results and the reported relative improvements appear to be statistically significant in most cases. A comparative analysis of the improvements for different concept integration strategies revealed that two separate effects lead to these improvements. A first effect that can be mainly attributed to multi-word expression detection and synonym conflation is achieved through the basic concept integration. A second effect building on this initial improvement is attributed to the use of the ontology structures for generalization through hypernym retrieval and integration.

Outlook The experiments that have been conducted show that the presented approach appears to be promising in most settings. However it has also become obvious that the results depend on the specific constellation of parameters. These include — most importantly — the choice of the appropriate ontology. Further research and experiments should investigate how the specific choice and setup of the used ontologies can lead to even better results and wether other concept extraction strategies lead to a further improvement in classification performance.

Further attention should also be paid to the setup of the classification algorithm as the general nature of AdaBoost would allow to integrate more advanced weak learners. Such weak learners might also exploit background knowledge even more directly.

Acknowledgements This research was partially supported by the European Commission under contract FP6-001765 aceMedia. The expressed content is the view of the authors but not necessarily the view of the aceMedia project as a whole.

References

1. R. C. Bodner and F. Song. Knowledge-Based Approaches to Query Expansion in Information Retrieval. In *Advances in Artificial Intelligence*. Springer, New York, NY, USA, 1996.
2. E. Bozsak et al. KAON – Towards a Large Scale Semantic Web. In *Proc. of the 3rd International Conference on E-Commerce and Web Technologies (EC-Web 2002)*, pages 304–313, Aix-en-Provence, France, 2002. LNCS 2455 Springer.
3. L. Cai and T. Hofmann. Text Categorization by Boosting Automatically Extracted Concepts. In *Proc. of the 26th Annual Int. ACM SIGIR Conference on Research and Development in Informaion Retrieval*, Toronto, Canada, 2003. ACM Press.
4. Y. Freund and R. E. Schapire. A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting. In *Second European Conference on Computational Learning Theory (EuroCOLT-95)*, pages 23–37, 1995.
5. S. Handschuh and S. Staab, editors. *Annotation for the Semantic Web*. IOS Press, 2003.
6. W. R. Hersh, C. Buckley, T. J. Leone, and D. H. Hickam. Ohsumed: An Interactive Retrieval Evealuation and new large Test Collection for Research. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. ACM Press, 1994.
7. A. Hotho, S. Staab, and G. Stumme. Wordnet improves Text Document Clustering. In *Proc. of the Semantic Web Workshop of the 26th Annual International ACM SIGIR Conference*, Toronto, Canada, 2003.
8. 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.
9. T. Joachims. Text Categorization with Support Vector Machines: Learning With Many Relevant Features. In *Proceedings of ECML-98*, 1998.
10. A. Kehagias, V. Petridis, V. G. Kaburlasos, and P. Fragkou. A Comparison of Word- and Sense-Based Text Categorization Using Several Classification Algorithms. *Journal of Intelligent Information Systems*, 21(3):227–247, 2000.
11. R. Meir and G. Rätsch. An Introduction to Boosting and Leveraging. In *Advanced Lectures on Machine Learning*, LNCS. Springer, Heidelberg, DE, 2003.
12. G. A. Miller, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: an On-Line Lexical Database. *International Journal of Lexicography*, 3(4):235–244, 1990.
13. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
14. G. Salton. *Automatic Text Processing*. Addison-Wesley Publishing Inc, Boston, MA, USA, 1989.
15. R. E. Schapire and Y. Singer. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2/3):135–168, 2000.
16. F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
17. B. B. Wang, R. I. Mckay, H. A. Abbass, and M. Barlow. A comparative study for domain ontology guided feature extraction. In *Proceedings of the 26th Australian Computer Science Conference (ACSC-2003)*, pages 69–78. Australian Computer Society, 2003.
18. Y. Yang. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1(1-2):69–90, 1999.
19. Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999.