# Text Classification by Boosting Weak Learners based on Terms and Concepts

Stephan Bloehdorn
University of Karlsruhe
Institute AIFB, Knowledge Management Group
D-76128 Karlsruhe, Germany
bloehdorn@aifb.uni-karlsruhe.de

Andreas Hotho
University of Kassel
Knowledge and Data Engineering Group
D-34121 Kassel, Germany
hotho@cs.uni-kassel.de

## Abstract

*Document representations for text classification are typically based on the classical Bag-Of-Words paradigm. This approach comes with deficiencies that motivate the integration of features on a higher semantic level than single words. In this paper we propose an enhancement of the classical document representation through concepts extracted from background knowledge. Boosting is used for actual classification. Experimental evaluations on two well known text corpora support our approach through consistent improvement of the results.*

## 1 Introduction

Most of the explicit knowledge assets of today's organizations consist of unstructured textual information in electronic form. Systems that contextualize information by automatically classifying text documents into predefined thematic classes help users to organize and exploit the ever growing amounts of textual information.

During the last decades, a large number of machine learning methods have been proposed for text classification tasks [8]. They are, however, typically built around the *Bag-of-Words model* known from information retrieval. In this representation, documents are considered to be bags of words, each term or term stem being an independent feature of it's own – typically represented through binary indicator variables, absolute frequencies or more elaborated measures like TFIDF [7]. Learning algorithms are thus 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 like *"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 "beef" and "pork" to their common hypernym "meat".

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 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. Evaluations on two well known text corpora show that our approach leads to consistent improvements.

## 2 Conceptual Document Representation

**Ontologies** The background knowledge we will exploit further on is encoded in a *core ontology*. For the purpose of this paper, we present only important parts of our more extensive ontology definition described in [2].

**Definition 2.1 (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 generalization hierarchy or taxonomy. The partial order* $<_C$ *relates the concepts in an ontology in form of specialization/generalization relationships.*

**Definition 2.2 (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* $Ref_C(s) := \{c \in C | (s, c) \in Ref_C\}$*.*

For the purpose of actual evaluation in the experiments, we have used two different resources, namely *WordNet* and the *MeSH Tree Structures Ontology*. Although not explicitly designed as an ontology, *WordNet*[1] largely fits into the ontology definitions given above. The WordNet database organizes 152,059 lexical index terms into a total of 115,424 so called *synonym sets (synsets)*, each of which represents an underlying concept and links these through semantic relations.

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)[2]. The ontology itself was ported into and accessed through the Karlsruhe Ontology and Semantic Web Infrastructure (KAON) infrastructure[3]. The ontology contains more than 22,000 concepts, each enriched with synonymous and quasi-synonymous language expressions.

**Concept Extraction from Texts** We have developed a process for extracting concepts from texts given a specific ontology. We shortly describe these steps in the following. The interested reader is referred to [1] for a more detailed description.

Due to the existence of multi-word expressions, the mapping of terms to concepts can not be accomplished by querying the lexicon directly for single words. 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. Our algorithm moves a window of a given length over the input text, analyzes the window content and either decreases the window size if the content can not be found in the lexicon or moves the window further to the next candidate expression. Querying the lexicon directly for any candidate expression in the window is likely to result in a large number of unnecessary queries. To increase efficiency and at the same time improve the concept retrieval quality we have incorporated a *syntactical analysis* step. By defining appropriate POS patterns (e.g. patterns for noun phrases) and matching the window content against these, expressions that will surely not symbolize concepts can be excluded in the first hand and different syntactic categories can be disambiguated.

Typically, the lexicon will not contain all inflected forms of its entries. If the lexicon interface is capable of performing the morphological transformations for base form reduction (e.g. in WordNet), queries can be processed directly. If the lexicon interface does not provide such functionalities, 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.

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. In our experiments, we have used three simple *Word Sense Disambiguation* strategies [5]:

1. The `'all'` strategy uses all possible concepts (no disambiguation).
2. The `'first'` strategy exploits WordNet's capability to return synsets ordered with respect to usage frequency by choosing the most frequent among several concepts.
3. The `'context'` strategy performs disambiguation based on a simple approach that also considers the overall document context for disambiguation as proposed in [5].

The last step in the process is about going from the specific concepts found in the text to more general concept representations. 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.

## 3 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. In this paper, we will concentrate on the well known AdaBoost algorithm [4] given on the next page and on simple indicator function decision stumps as base learners. These latter have the form:

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

where $c \in \{-1, 1\}$. These decision stumps take 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. $c = -1$.

## 4 Experiments

**Evaluation Metrics** We have used a standard set of evaluation metrics commonly used in IR to assess the performance of our approach, namely the *classification error*, *precision*, *recall*, the $F_1$ *measure* and break-even point (BEP). To average evaluation results of binary classifications on the per-class level, two conventional methods exist. The *macro-averaged* figures are meant to be averages over the individual results of the different classes while *micro-averaged* fig-

[1] see `http://www.cogsci.princeton.edu/~wn/`
[2] see `http://www.nlm.nih.gov/mesh/`
[3] see `http://kaon.semanticweb.org/`

**Algorithm 1** 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 ensuring 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:
        $\hat{f}(x) = \text{sign}\left(\hat{f}_{soft}(x)\right) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$    (4)
```

ures are calculated over all individual documents. Refer to [8] for a detailed description of these measures.

**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.

In the first stage of the experiment, term stems[4] and WordNet concepts were extracted as features from the documents in the training and test corpus. As a result, 17,525 distinct term stems and – depending on the chosen disambiguation strategy and maximal generalization (hypernym) distance – 10,259 to 27,236 distinct concept features. 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.

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.

| | | 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.**

**Evaluation on the OHSUMED Corpus** A second series of experiments was conducted using the 1987 portion of the OHSUMED collection[5] consisting of 54,708 titles and abstracts from medical journals indexed with MeSH descriptors. About two thirds, 36,369 documents, were randomly selected as training documents, the remaining 18,341 documents were used for testing. For term stems, a total number of 38,047 distinct features could be identified. WordNet and the MeSH Tree Structures Ontology were used to extract conceptual features. With WordNet, all different disambiguation strategies were used resulting in 16,442 to 34,529 synset features. For the MeSH Tree Structures Ontology, only the "all" strategy was used, resulting in 11,572 to 13,663 MeSH concept features. Again, binary classification was performed with AdaBoost on the top 50 categories where the number of boosting iterations was set to 1000 rounds. Different runs of the classification stage were performed based on the different features, leading to often substantially different results.

Table 2 summarizes the macro- and micro-averaged results. Again, the general finding is that complementing the term stem representation with conceptual features significantly improves classification performance. The relative improvements for the $F_1$ scores compared to the term stem baseline range from 2.40 % to 6.98 % on the macro

[4] In this and in the next experiment term stem extraction comprises the removal of the standard stopwords for English defined in the SMART stopword list and stemming using the porter stemming algorithm.

[5] see `http://trec.nist.gov/data/t9_filtering.html`

level and from 1.96 % to 6.53 % on the micro level. The relative improvements achieved on OHSUMED are generally higher than those achieved on the Reuters-21578 corpus. This makes intuitively sense as the documents in the OHSUMED corpus are taken from the medical domain and are therefore typically suffering from the problems described in section 1, especially synonymous terms and multi-word expressions. 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.

| | | **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 OHSUMED.**

## 5 Related Work

To date, the work on integrating semantic background knowledge into text classification or other related tasks is quite scattered and has often lead to disappointing results. For example, a comparison of a number of approaches based on word-sense document representations reported in [6] ends with the conclusion of the authors that *"the use of word senses does not result in any significant categorization improvement"*.

Improvements resulting from feature representations based on ontological concepts were reported in text clustering settings [5]. Very good results with a feature representation mixed of terms and "concepts" computed statistically by means of 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.

## 6 Conclusions

In this paper, we have proposed an approach to incorporate concepts from background knowledge into document representations for text document classification. AdaBoost, was used for actual classifications. Experiments on the Reuters and OHSUMED datasets clearly show that the integration of concepts into the feature representation improves classification results. The scores achieved are highly competitive with other published results. A series of statistical significance tests we have ommitted in full detail due to space restrictions indicates that the reported relative improvements can be assessed significant in most cases.

A comparative analysis of the improvements for different concept integration strategies revealed that these are due to two separate effects. Firstly, some improvements can be attributed to the detection of multi-word expressions and conflation of synonyms achieved through basic concept integration. Building on this initial improvement, further improvements can be achieved by generalization through superconcept retrieval and integration.

**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] S. Bloehdorn and A. Hotho. Boosting for Text Classification with Semantic Features. In *Proceedings of the MSW 2004 Workshop at the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* , Seattle, WA, USA, 2004.

[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)*, Aix-en-Provence, France, 2002.

[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.

[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)*, Barcelona, Spain, 1995.

[5] A. Hotho, S. Staab, and G. Stumme. Wordnet improves Text Document Clustering. In *Proceedings of the Semantic Web Workshop at the 26th Annual International ACM SIGIR Conference*, Toronto, Canada, 2003.

[6] 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.

[7] G. Salton. *Automatic Text Processing*. Addison-Wesley Publishing Inc, Boston, MA, USA, 1989.

[8] F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.