Text Categorization from Category Name via Lexical Reference

Libby Barak
Department of Computer Science
University of Toronto
Toronto, Canada M5S 1A4
libbyb@cs.toronto.edu

Ido Dagan and **Eyal Shnarch**
Department of Computer Science
Bar-Ilan University
Ramat-Gan 52900, Israel
{dagan, shey}@cs.biu.ac.il

Abstract

Requiring only category names as user input is a highly attractive, yet hardly explored, setting for text categorization. Earlier bootstrapping results relied on similarity in LSA space, which captures rather coarse contextual similarity. We suggest improving this scheme by identifying concrete references to the category name's meaning, obtaining a special variant of lexical expansion.

1 Introduction

Topical Text Categorization (TC), the task of classifying documents by pre-defined topics, is most commonly addressed as a supervised learning task. However, the supervised setting requires a substantial amount of manually labeled documents, which is often impractical in real-life settings.

Keyword-based TC methods (see Section 2) aim at a more practical setting. Each category is represented by a list of characteristic keywords, which should capture the category meaning. Classification is then based on measuring similarity between the category keywords and the classified documents, typically followed by a bootstrapping step. The manual effort is thus reduced to providing a keyword list per category, which was partly automated in some works through clustering.

The keyword-based approach still requires non-negligible manual work in creating a representative keyword list per category. (Gliozzo et al., 2005) succeeded eliminating this requirement by using the category name alone as the initial keyword, yet obtaining superior performance within the keyword-based approach. This was achieved by measuring similarity between category names and documents in *Latent Semantic* space (LSA), which implicitly captures contextual similarities for the category name through unsupervised dimensionality reduction. Requiring only category names as user input seems very attractive, particularly when labeled training data is too costly while modest performance (relative to supervised methods) is still useful.

The goal of our research is to further improve the scheme of text categorization from category name, which was hardly explored in prior work. When analyzing the behavior of the LSA representation of (Gliozzo et al., 2005) we noticed that it captures two types of similarities between the category name and document terms. One type regards words which refer specifically to the category name's meaning, such as *pitcher* for the category `Baseball`. However, typical context words for the category which do not necessarily imply its specific meaning, like *stadium*, also come up as similar to *baseball* in LSA space. This limits the method's precision, due to false-positive classifications of contextually-related documents that do not discuss the specific category topic (such as other sports documents wrongly classified to `Baseball`). This behavior is quite typical for query expansion methods, which expand a query with contextually correlated terms.

We propose a novel scheme that models separately these two types of similarity. For one, it identifies words that are likely to refer *specifically* to the category name's meaning (Glickman et al., 2006), based on certain relations in WordNet and

Proceedings of NAACL HLT 2009: Short Papers, pages 33–36,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

Wikipedia. In tandem, we assess the general contextual fit of the category topic using an LSA model, to overcome lexical ambiguity and passing references. The evaluations show that tracing lexical references indeed increases classification precision, which in turn improves the eventual classifier obtained through bootstrapping.

2 Background: Keyword-based Text Categorization

The majority of keyword-based TC methods fit the general bootstrapping scheme outlined in Figure 1, which is cast in terms of a vector-space model. The simplest version for step 1 is manual generation of the keyword lists (McCallum and Nigam, 1999). (Ko and Seo, 2004; Liu et al., 2004) partly automated this step, using clustering to generate candidate keywords. These methods employed a standard term-space representation in step 2.

As described in Section 1, the keyword list in (Gliozzo et al., 2005) consisted of the category name alone. This was accompanied by representing the category names and documents (step 2) in LSA space, obtained through cooccurrence-based dimensionality reduction. In this space, words that tend to cooccur together, or occur in similar contexts, are represented by similar vectors. Thus, vector similarity in LSA space (in step 3) captures implicitly the similarity between the category name and contextually related words within the classified documents.

Step 3 yields an initial similarity-based classification that assigns a single (most similar) category to each document, with $Sim(c, d)$ typically being the cosine between the corresponding vectors. This classification is used, in the subsequent bootstrapping step, to train a standard supervised classifier (either single- or multi-class), yielding the eventual classifier for the category set.

3 Integrating Reference and Context

Our goal is to augment the coarse contextual similarity measurement in earlier work with the identification of concrete references to the category name's meaning. We were mostly inspired by (Glickman et al., 2006), which coined the term *lexical reference* to denote concrete references in text to the specific meaning of a given term. They further showed that an entailing text (in the textual entailment setting) typically includes a concrete reference to each term in the entailed statement. Analogously, we assume that a relevant document for a category typically includes concrete terms that refer *specifically* to the category name's meaning.

```
Input: set of categories and unlabeled documents
Output: a classifier
1. Acquire a keyword list per category
2. Represent each category $c$ and document $d$
   as vectors in a common space
3. For each document $d$
     $Cat_{Sim}(d) = argmax_c(Sim(c, d))$
4. Train a supervised classifier on step (3) output
```

Figure 1: Keyword-based categorization scheme

Category name	WordNet	Wikipedia
Cryptography	*decipher*	*digital signature*
Medicine	*cardiology*	*biofeedback, homeopathy*
Macintosh		*Apple Mac, Mac*
Motorcycle	*bike, cycle*	*Honda XR600*

Table 1: Referring terms from WordNet and Wikipedia

We thus extend the scheme in Figure 1 by creating two vectors per category (in steps 1 and 2): a *reference vector* $\vec{c}_{ref}$ in term space, consisting of referring terms for the category name; and a *context vector* $\vec{c}_{con}$, representing the category name in LSA space, as in (Gliozzo et al., 2005). Step 3 then computes a combined similarity score for categories and documents based on the two vectors.

3.1 References to category names

Referring terms are collected from WordNet and Wikipedia, by utilizing relations that are likely to correspond to lexical reference. Table 1 illustrates that WordNet provides mostly referring terms of general terminology while Wikipedia provides more specific terms. While these resources were used previously for text categorization, it was mostly for enhancing document representation in supervised settings, e.g. (Rodríguez et al., 2000).

WordNet. Referring terms were found in WordNet starting from relevant senses of the category name and transitively following relation types that correspond to lexical reference. To that end, we specified for each category name those senses which fit the category's meaning, such as the *outer space* sense for the category `Space`.[1]

A category name sense is first expanded by its synonyms and derivations, all of which are then expanded by their hyponyms. When a term has no hyponyms it is expanded by its meronyms instead, since we observed that in such cases they often specify unique components that imply the holonym's meaning, such as *Egypt* for *Middle East*. However, when a term is not a leaf in the hyponymy hierarchy then its meronyms often refer to generic sub-parts, such as *door* for *car*. Finally, the hyponyms and meronyms are expanded by their derivations. As a common heuristic, we considered only the most frequent senses (top 4) of referring terms, avoiding low-ranked (rare) senses which are likely to introduce noise.

Wikipedia. We utilized a subset of a lexical reference resource extracted from Wikipedia (anonymous reference). For each category name we extracted referring terms of two types, capturing hyponyms and synonyms. Terms of the first type are Wikipedia page titles for which the first definition sentence includes a syntactic "is-a" pattern whose complement is the category name, such as *Chevrolet* for the category `Autos`. Terms of the second type are extracted from Wikipedia's redirect links, which capture synonyms such as *x11* for `Windows-X`.

The reference vector $\vec{c}_{ref}$ for a category consists of the category name and all its referring terms, equally weighted. The corresponding similarity function is $Sim_{ref}(c,d) = \cos(\vec{c}_{ref}, \vec{d}_{term})$, where $\vec{d}_{term}$ is the document vector in term space.

3.2 Incorporating context similarity

Our key motivation is to utilize Sim_{ref} as the basis for classification in step 3 (Figure 1). However, this may yield false positive classifications in two cases: (a) inappropriate sense of an ambiguous referring term, e.g., the narcotic sense of *drug* should not yield classification to `Medicine`; (b) a passing reference, e.g., an analogy to *cars* in a software document, should not yield classification to `Autos`.

In both these cases the overall context in the document is expected to be atypical for the triggered category. We therefore measure the contextual similarity between a category c and a document d utilizing LSA space, replicating the method in (Gliozzo et al., 2005): $\vec{c}_{con}$ and $\vec{d}_{LSA}$ are taken as the LSA vectors of the category name and the document, respectively, yielding $Sim_{con}(c,d) = \cos(\vec{c}_{con}, \vec{d}_{LSA})$.[2]

The overall similarity score of step 3 is defined as $Sim(c,d) = Sim_{ref}(c,d) \cdot Sim_{con}(c,d)$. This formula fulfils the requirement of finding at least one referring term in the document; otherwise $Sim_{ref}(c,d)$ would be zero. $Sim_{con}(c,d)$ is computed in the reduced LSA space and is thus practically non-zero, and would downgrade $Sim(c,d)$ when there is low contextual similarity between the category name and the document. Documents for which $Sim(c,d) = 0$ for all categories are omitted.

4 Results and Conclusions

We tested our method on the two corpora used in (Gliozzo et al., 2005): 20-NewsGroups, classified by a single-class scheme (single category per document), and Reuters-10 [3], of a multi-class scheme. As in their work, non-standard category names were adjusted, such as *Foreign exchange* for `Money-fx`.

4.1 Initial classification

Table 2 presents the results of the initial classification (step 3). The first 4 lines refer to classification based on Sim_{ref} alone. As a baseline, including only the category name in the reference vector (*CatName*) yields particularly low recall. Expansion by *WordNet* is notably more powerful than by the automatically extracted *Wikipedia* resource; still, the latter consistently provides a small marginal improvement when using both resources (*Reference*), indicating their complementary nature.

As we hypothesized, the *Reference* model achieves much better precision than the *Context* model from (Gliozzo et al., 2005) alone (Sim_{con}). For 20-NewsGroups the recall of *Reference* is limited, due to partial coverage of our current expansion

[1] We assume that it is reasonable to specify relevant senses as part of the typically manual process of defining the set of categories and their names. Otherwise, when expanding names through all their senses F1-score dropped by about 2%.

[2] The original method includes a Gaussian Mixture rescaling step for Sim_{con}, which wasn't found helpful when combined with Sim_{ref} (as specified next).

[3] 10 most frequent categories in *Reuters-21578*

	Reuters-10			20 Newsgroups		
Method	R	P	F1	R	P	F1
CatName	0.22	0.67	0.33	0.19	0.55	0.28
WordNet	0.67	0.78	0.72	0.29	0.56	0.38
Wikipedia	0.24	0.68	0.35	0.22	0.57	0.31
Reference	0.69	0.80	0.74	0.31	0.57	0.40
Context	0.59	0.64	0.61	**0.46**	0.46	**0.46**
Combined	**0.71**	**0.82**	**0.76**	0.32	**0.58**	0.41

Table 2: Initial categorization results (step 3)

Method	Feature Set	Reuters-10			20 NG
		R	P	F1	F1
Reference	TF-IDF	0.91	0.50	0.65	0.51
	LSA	0.89	0.67	0.76	**0.56**
Context	TF-IDF	0.84	0.48	0.61	0.48
	LSA	0.73	0.56	0.63	0.44
Combined	TF-IDF	**0.92**	0.50	0.65	0.52
	LSA	0.89	**0.71**	**0.79**	**0.56**

Table 3: Final bootstrapping results (step 4)

resources, yielding a lower F1. Yet, its higher precision pays off for the bootstrapping step (Section 4.2). Finally, when the two models are *Combined* a small precision improvement is observed.

4.2 Final bootstrapping results

The output of step 3 was fed as standard training for a binary SVM classifier for each category (step 4). We used the default setting for SVM-light, apart from the *j* parameter which was set to the number of categories in each data set, as suggested by (Morik et al., 1999). For Reuters-10, classification was determined independently by the classifier of each category, allowing multiple classes per document. For 20-NewsGroups, the category which yielded the highest classification score was chosen (one-versus-all), fitting the single-class setting. We experimented with two document representations for the supervised step: either as vectors in tf-idf weighted term space or as vectors in LSA space.

Table 3 shows the final classification results.[4] First, we observe that for the noisy bootstrapping training data LSA document representation is usually preferred. Most importantly, our *Reference* and *Combined* models clearly improve over the earlier *Context*. Combining reference and context yields some improvement for Reuters-10, but not for 20-NewsGroups. We noticed though that the actual accuracy of our method on 20-NewsGroups is notably higher than measured relative to the gold standard, due to its single-class scheme: in many cases, a document should truly belong to more than one category while that chosen by our algorithm was counted as false positive. Future research is proposed to increase the method's recall via broader coverage lexical reference resources, and to improve its precision through better context models than LSA, which was found rather noisy for quite a few categories.

To conclude, the results support our main contribution – the benefit of identifying *referring terms* for the category name over using noisier context models alone. Overall, our work highlights the potential of text categorization from category names when labeled training sets are not available, and indicates important directions for further research.

Acknowledgments

The authors would like to thank Carlo Strapparava and Alfio Gliozzo for valuable discussions. This work was partially supported by the NEGEV project (www.negev-initiative.org).

References

O. Glickman, E. Shnarch, and I. Dagan. 2006. Lexical reference: a semantic matching subtask. In *EMNLP*.

A. Gliozzo, C. Strapparava, and I. Dagan. 2005. Investigating unsupervised learning for text categorization bootstrapping. In *Proc. of HLT/EMNLP*.

Y. Ko and J. Seo. 2004. Learning with unlabeled data for text categorization using bootstrapping and feature projection techniques. In *Proc. of ACL*.

B. Liu, X. Li, W. S. Lee, and P. S. Yu. 2004. Text classification by labeling words. In *Proc. of AAAI*.

A. McCallum and K. Nigam. 1999. Text classification by bootstrapping with keywords, EM and shrinkage. In *ACL Workshop for Unsupervised Learning in NLP*.

K. Morik, P. Brockhausen, and T. Joachims. 1999. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proc. of the 16th Int'l Conf. on Machine Learning*.

M. d. B. Rodríguez, J. M. Gómez-Hidalgo, and B. Díaz-Agudo, 2000. *Using WordNet to complement training information in text categorization*, volume 189 of Current Issues in Linguistic Theory, pages 353–364.

[4] Notice that P=R=F1 when *all* documents are classified to a single class, as in step 4 for 20-NewsGroups, while in step 3 some documents are not classified, yielding distinct P/R/F1.