Journal of Machine Learning Research X (YYYY) 1-1 Submitted 02/06; Published

Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization*

Evgeniy Gabrilovich† GABR@YAHOO-INC.COM
Shaul Markovitch SHAULM@CS.TECHNION.AC.IL
Department of Computer Science
Technion—Israel Institute of Technology, 32000 Haifa, Israel

Editor: Andrew McCallum

Abstract

Most existing methods for text categorization employ induction algorithms that use the words appearing in the training documents as features. While they perform well in many categorization tasks, these methods are inherently limited when faced with more complicated tasks where external knowledge is essential. Recently, there have been efforts to augment these basic features with external knowledge, including semi-supervised learning and transfer learning. In this work, we present a new framework for automatic acquisition of world knowledge and methods for incorporating it into the text categorization process. Our approach enhances machine learning algorithms with features generated from domain-specific and common-sense knowledge. This knowledge is represented by ontologies that contain hundreds of thousands of concepts, further enriched through controlled Web crawling. Prior to text categorization, a feature generator analyzes the documents and maps them onto appropriate ontology concepts that augment the bag of words used in simple supervised learning. Feature generation is accomplished through contextual analysis of document text, thus implicitly performing word sense disambiguation. Coupled with the ability to generalize concepts using the ontology, this approach addresses two significant problems in natural language processing—synonymy and polysemy. Categorizing documents with the aid of knowledge-based features leverages information that cannot be deduced from the training documents alone. We applied our methodology using the Open Directory Project, the largest existing Web directory built by over 70,000 human editors. Experimental results over a range of datasets confirm improved performance compared to the bag of words document representation.

Keywords: Feature generation, text classification, background knowledge.

1. Introduction

Text categorization deals with assigning category labels to textual documents. Categories come from a fixed set of labels (possibly organized in a hierarchy) and each document may be assigned one or more categories. Text categorization systems are useful in a wide variety

[*] A preliminary version of this paper appeared in the Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, UK, August 2005 (Gabrilovich and Markovitch, 2005).

[†] Current address: Yahoo! Research, 2821 Mission College Blvd, Santa Clara, CA 95054, USA.

of tasks, such as routing news and e-mail to appropriate corporate desks, identifying junk email, or correctly handling intelligence reports.

The majority of existing text classification systems use various induction techniques, such as support vector machines, k-nearest neighbor algorithm, and neural networks. The features commonly used are the individual words appearing in the training documents (while their order within the document is ignored). The value of a feature for a particular document is usually its occurrence frequency normalized by its occurrence frequency within the whole collection of documents. This representation scheme treats each document as a bag of the words it contains, and is therefore known as the *bag of words* (BOW) approach (Salton and McGill, 1983).

The bag of words method is very effective in easy to medium difficulty categorization tasks where the category of a document can be identified by several easily distinguishable keywords. There are, however, two major weaknesses to the BOW representation scheme that limit its usefulness for more demanding categorization tasks. The first one stems from representing a document as a *bag*, thus ignoring the order of words appearance. This limits the possibility of handling structures that are based on more than one word, and also limits the possibility of disambiguating words based on their context.

The second weakness is the usage of only words that are explicitly mentioned in the training documents, without any knowledge about them. Because this approach cannot generalize over words, words in the testing document that never appeared in the training set are necessarily ignored. Nor can synonymous words that appear infrequently in training documents be used to infer a more general principle that covers several cases.

There have been a number of efforts to extend the basic BOW approach. Several studies augmented the bag of words with n-grams (Caropreso et al., 2001; Peng and Shuurmans, 2003; Mladenic, 1998b; Raskutti et al., 2001) or statistical language models (Peng et al., 2004). Others used linguistically motivated features based on syntactic information, such as that available from part-of-speech tagging or shallow parsing (Sable et al., 2002; Basili et al., 2000). Additional studies researched the use of word clustering (Baker and McCallum, 1998; Bekkerman, 2003; Dhillon et al., 2003), as well as dimensionality reduction techniques such as LSA (Deerwester et al., 1990; Hull, 1994; Zelikovitz and Hirsh, 2001; Cai and Hofmann, 2003).

More recently, there have been a number of efforts to add outside knowledge to supervised machine learning techniques. Transfer learning approaches (Bennett et al., 2003; Do and Ng, 2005; Sutton and McCallum, 1998; Raina et al., 2006) leverage information from different but related learning tasks. Pseudo-relevance feedback (Ruthven and Lalmas, 2003) uses information from the top-ranked documents, which are assumed to be relevant to the query; for example, characteristic terms from such documents may be used for query expansion (Xu and Croft, 1996). Recent studies on semi-supervised methods (Goldberg and Zhu, 2006; Ando and Zhang, 2005a,b; Blei et al., 2003; Nigam et al., 2000; Joachims, 1999b) infer information from unlabeled data, which is often available in much larger amounts than labeled data.

We argue that in order to perform text categorization well, the computer needs access to much more extensive and deep knowledge. Over a decade ago, Lenat and Feigenbaum (1990) formulated the *knowledge principle*, which postulated that “If a program is to perform a complex task well, it must know a great deal about the world it operates in.” Text categorization is certainly a complex task. While the basic approaches are able to identify commonalities between documents based on word identity, and more advanced approaches can recognize synonyms, there are cases where identifying commonality between documents requires recognition of more elaborated semantic relations between terms.

For illustration, consider document #15264 in Reuters-21578, which is one of the most frequently used datasets in text categorization research. This document discusses a joint mining venture by a consortium of companies, and belongs to the category "copper." However, this fairly long document mentions only briefly that the aim of this venture is mining copper; rather, its main focus is on the mutual share holdings of the companies involved (Teck Corporation, Cominco, and Lornex Mining), as well as other mining activities of the consortium. Consequently, the three very different text classifiers that we used (SVM, KNN and C4.5) failed to classify the document correctly. This comes as no surprise—"copper" is a fairly small category, and none of these companies, nor the location of the venture (Highland Valley in British Columbia, Canada) is ever mentioned in the training set for this category.

We argue that this need not be the case. When a Reuters editor originally handled this document, she most likely knew quite a lot about the business of these companies, and easily assigned the document to the category "copper." It is this kind of knowledge that we would like machine learning algorithms to have access to.

In this paper we introduce a method for enhancing machine learning algorithms with a large volume of knowledge extracted from available human-generated repositories. Our method capitalizes on the power of existing induction techniques while enriching the language of representation, namely, exploring new feature spaces. Prior to text categorization, we employ a *feature generator* that uses common-sense and domain-specific knowledge to enrich the bag of words with new, more informative and discriminating features. Feature generation is performed automatically, using machine-readable hierarchical repositories of knowledge. Many sources of world knowledge have become available in recent years, thanks to rapid advances in information processing, and Internet proliferation in particular. Examples of general purpose knowledge bases include the Open Directory Project (ODP), Yahoo! Web Directory, and the Wikipedia encyclopedia.

It is interesting to juxtapose our method with above-mentioned alternative approaches that augment the training set of documents with external knowledge. Semi-supervised learning uses unlabeled data to gather additional features beyond those originally available in the input. Transfer learning involves pairs of related learning tasks, so that features constructed while solving one problem can then also be used for solving another problem. On the other hand, the methods we propose in this paper build new features using knowledge explicitly cataloged by humans, which comes in the form of concepts that correspond to the nodes of the Open Directory.

In this paper we use the ODP as a source of background knowledge. The Open Directory catalogs millions of Web sites in a rich hierarchy of 600,000 categories, and represents the collective knowledge of over 70,000 volunteer editors. Thus, in the above example, the feature generator "knows" that the companies mentioned are in the mining business, and that Highland Valley happens to host a copper mine. This information is available in Web pages that discuss the companies and their operations, and are cataloged in corresponding ODP categories such as Mining_and_Drilling and Metals. Similarly, Web pages about Highland Valley are cataloged under Regional/North_America/Canada/British_Columbia. To amass this information, we crawl the URLs cataloged in the ODP, thus effectively multiplying the amount of knowledge available many times over. Armed with this knowledge, the feature generator constructs new features that denote these ODP categories, and adds them to the bag of words. The augmented feature space provides text classifiers with a cornucopia of additional information. Indeed, our implementation of the proposed methodology classifies this document correctly. It is essential to mention that this entire scheme works automatically. Given an existing knowledge hierarchy (ODP in this case), the feature generator examines documents and enriches their representation in a completely mechanical way.

The contributions of this paper are threefold. First, we propose a framework and a collection of algorithms that perform feature generation using very large-scale repositories of human knowledge. Second, we propose a novel kind of contextual analysis performed during feature generation, which views the document text as a sequence of local contexts, and performs implicit word sense disambiguation. Finally, we describe a way to further enhance existing knowledge bases by several orders of magnitude by crawling the World Wide Web. Performing feature generation using external knowledge effectively capitalizes on human knowledge (as encoded by the editors of the Open Directory), leveraging information that cannot be deduced solely from the texts being classified. As we show in Section 5, our approach performs markedly better than the bag of words method.

We believe that this research is only one step towards computerized use of large-scale structured repositories of human knowledge. In our future work, we plan to study possible uses of other knowledge repositories in addition to the Open Directory. We also intend to apply the feature generation methodology to additional natural language processing tasks, as well as to study its applicability beyond text processing. It would also be very interesting to compare the results of the feature generation methodology presented in this paper to other techniques that use unlabeled data, such as semi-supervised and transfer learning; this comparison is also left to future work.

The rest of the paper is organized as follows. In Section 2 we analyze the limitations of the BOW approach. Section 3 describes how our feature generation methodology uses repositories of human knowledge to overcome these limitations. Section 4 instantiates this methodology with a particular knowledge resource, the Open Directory Project. In Section 5 we report the results of evaluating the proposed methodology empirically on a variety of test collections, and outline the implementation details of our system. In Section 6 we discuss our methodology in the context of prior work and related literature. Section 7 concludes the paper and outlines directions for future research.

2. Problems in the Bag of Words Approach

Since the majority of existing text categorization systems employ the bag of words approach to represent documents, we begin by analyzing typical problems and limitations of this method.

1. Words that appear in *testing* documents but not in *training* documents are completely ignored by the basic BOW approach that does not use external data to compensate for such vocabulary mismatch. Since the classification model is built with a subset

of words that appear in the training documents, words that do not appear there are excluded by definition. Lacking the ability to analyze such words, the system may overlook important parts of the document being classified.

Example: Document #15264 from Reuters-21578 described in the Introduction presents a perfect example of this limitation. This document describes a copper-mining venture formed by a group of companies, whose names are not mentioned even once in the training set, and are thus ignored by the classification model.

2. Words that appear infrequently in the training set, or appear just once, are mostly ignored even if they are essential for proper classification. It often happens that human annotators assign a document to a certain category based on some notion briefly mentioned in the document. If the words that describe this notion do not appear with sufficient frequency elsewhere in the training set, then the system will overlook the real reason for this document's annotation. Consequently, it will either come up with some spurious association between the actual category and unrelated words or ignore this document as a training example altogether.

   Example: Suppose we have a collection of pharmaceutical documents and are trying to learn the concept of antibiotics. If a particular training document describes the results of a clinical trial for a new antibiotic drug, and mentions it only by a brand name that does not appear elsewhere in the training set, the system will likely miss an important piece of evidence.

3. The problem described in the previous item can manifest itself in a more extreme way. Suppose we have a group of related words, where each word appears only a few times in the collection, and few documents contain more than one word of the group. As a result, the connection between these words remains implicit and cannot be learned without resorting to external knowledge. External knowledge, however, allows us to determine that certain words are related. Furthermore, we can use the generalization ability of hierarchical knowledge organization to establish that the words correspond to specific instances of the same general notion.

   Example: Consider a collection of clinical narrative reports on administering various antibiotic drugs. Since such reports are circulated among medical professionals, they are likely to refer to specific drugs by name, while omitting the knowledge already shared by the target audience. Hence, the reports will likely not explain that each drug is actually an antibiotic. In the absence of this vital piece of knowledge, the BOW approach can easily fail to learn the notion shared by the reports.

   Speaking more generally, we observe that a critical limitation of the BOW approach lies in its ignorance of the connections between the words. Thus, even more difficult than the problem described in the previous item, is the one where we have several related phrases or longer contexts, while the connection between them is not stated in any single document.

   Example: Consider again a collection of clinical reports, which are inherently rich in diverse medical terminology. Often, each report describes the case of a single patient. Thus, without extensive medical knowledge it would be nearly impossible to learn that

Lown-Ganong-Levine Syndrome and Wolff-Parkinson-White Syndrome are different kinds of arrhythmia, while Crigler-Najjar Syndrome and Gilbert Syndrome are two kinds of liver diseases.

4. Because contextual adjacency of words is not taken into account by the BOW approach, word sense disambiguation can only be performed at the level of entire documents, rather than at much more linguistically plausible levels of a single sentence or paragraph.

   Example: As an extreme example of this limitation, consider a document about the Jaguar company establishing a conservation trust to protect its namesake animal (`http://www.jaguarusa.com/us/en/company/news_events/archive/Jaguar_Conservation_trust_longcopy.htm`). This fairly long document is devoted mainly to the preservation of wildlife, while briefly covering the history of the car manufacturer in its last paragraph. Taken as a single bag of words, the document will likely be classified as strongly related to jaguar the animal, while the cursory mention of Jaguar the company will likely be ignored.

Some of these limitations are due to data sparsity—after all, if we had infinite amounts of text on every imaginable topic, the bag of words would perform much better. Many studies in machine learning and natural language processing addressed the sparsity problem. Approaches like smoothing (Chen and Goodman, 1996) allocate some probability mass for unseen events and thus eliminate zero probabilities. These approaches facilitate methods that are sensitive to zero probabilities (e.g., Naive Bayes), but essentially do not use any external knowledge. More elaborate techniques such as transfer learning (Bennett et al., 2003; Do and Ng, 2005; Sutton and McCallum, 1998; Raina et al., 2006) and semi-supervised learning (Goldberg and Zhu, 2006; Ando and Zhang, 2005a,b; Blei et al., 2003; Nigam et al., 2000; Joachims, 1999b), leverage cooccurrence information from similar learning tasks or from unlabeled data. Other studies that addressed the sparsity problem include using the EM algorithm with unlabeled data (Nigam et al., 2006, 2000), latent semantic kernels (Cristianini et al., 2002), transductive inference (Joachims, 1999b), and generalized vector space model (Wong et al., 1985).

Humans avoid these limitations due to their extensive world knowledge, as well as their ability to understand the words in context rather than just view them as an unordered bag. Our approach that uses structured background knowledge is somewhat reminiscent of explanation-based learning (Mitchell et al., 1986; Dejong and Mooney, 1986), where generalizations of previously seen examples are reused in future problem solving tasks, thus mimicking humans' ability to learn from a single example. Later in the paper we show how the above problems and limitations can be resolved through the use of knowledge-based feature generation.

3. Feature Generation Methodology

Having presented the problems with the BOW approach in the previous section, we continue by defining the guidelines for building a feature generation framework that will address and alleviate these problems using repositories of human knowledge.

3.1 Overview

The proposed methodology allows principled and uniform integration of one or more sources of external knowledge to construct new features. These knowledge sources define a collection of concepts that are assigned to documents to qualify their text. In the preprocessing step, we build a feature generator capable of representing documents in the space of these concepts. The feature generator is then invoked prior to text categorization to assign each document with a number of relevant concepts. Subsequently, these concepts give rise to a set of constructed features that provide background knowledge about the document's content. The constructed features can then be used either in conjunction with or in place of the original bag of words. The resulting set undergoes feature selection, and the most discriminative features are retained for document representation. Finally, we use traditional text categorization techniques to learn a text categorizer in the new feature space.

3.2 Requirements on Suitable Knowledge Repositories

We impose the following requirements on knowledge repositories for feature generation:

1. The repository contains a collection of *concepts* organized in a hierarchical tree structure, where edges represent the "is-a" relationship. Each hierarchy node is labeled with a concept, which is more general than those of its children. Although in principle we could perform feature generation with a flat set of concepts, using a hierarchical ontology allows us to perform powerful generalizations. Optionally, a concept may be accompanied by a brief textual description.

   Formally, let KR be a knowledge repository that contains concepts $C = \{c_0, \ldots, c_n\}$. Let c_0 be the *root node*, which is more general than any other node. Let $Parent(c_i)$ be a function that uniquely associates a node with its parent in the hierarchy, whereas $Parent(c_0)$ is undefined. Let $Children(c_i)$ be a function that associates a node with a set of its children, where for leaf nodes $Children(c_j) = \emptyset$. When concept c_i is more general than another concept c_j, we denote this by $c_i \sqsubseteq c_j$; this happens when $c_j \in Children^*(c_i)$, where $Children^*$ denotes the recursive application of the function (obviously, $\forall j > 0 : c_0 \sqsubseteq c_j$). If additional textual description is available for a concept, it is denoted by $Description(c_i)$; otherwise this function returns an empty set of words.

2. There is a collection of texts associated with each concept. The feature generator uses these texts to learn the definition and scope of the concept, in order to be able to assign it to relevant documents. We refer to these texts as *textual objects*, and denote the set of such objects associated with concept c_i as $T_i = \{t_{i,1}, \ldots, t_{i,m_i}\}$.

Let W be a set of words. Our goal is to build a mapping function $f : W^* \to 2^C$. We propose building the mapping function using text categorization techniques. This is a very natural thing to do, as text categorization is all about assigning documents or parts thereof to a predefined set of categories (concepts in our case). One way to do so is to use a binary learning algorithm $L(Pos, Neg)$ to build a set of n binary classifiers, $f_1, \ldots, f_n$, such that $f_i : W^* \to \{0, 1\}$. This way, individual classifiers are built using the chosen learning algorithm: $f_i = L(T_i, \bigcup_{1 \le j \le n, j \ne i} T_j)$. Another way to build such a mapping function is to devise a hierarchical text classifier that takes advantage of the hierarchical organization of categories. In this paper, we use a simpler approach of building a single classifier that simultaneously considers all categories for each input sequence of words.

We believe that the above requirements are not overly restrictive. Indeed, there are quite a few sources of common-sense and domain-specific knowledge that satisfy these requirements. We list below several notable examples.

- Internet directories such as the Yahoo Web Directory (`http://dir.yahoo.com`), the Open Directory Project (`http://www.dmoz.org`) and the LookSmart directory (`http://search.looksmart.com/p/browse`) catalog huge numbers of URLs organized in an elaborate hierarchy. The Web sites pointed at by these URLs can be crawled to gather a wealth of information about each directory node. Here each directory node defines a concept, and crawling the Web sites cataloged under the node provides a collection of textual objects for that node.

- The Medical Subject Headings (MeSH) taxonomy (MeSH, 2003), which defines over 18,000 categories and is cross-linked with the MEDLINE database of medical articles, is a notable example of a domain-specific knowledge base. Here the hierarchy nodes again induce a set of concepts. The MEDLINE links mean that MeSH nodes can be easily associated with numerous scientific articles that are highly relevant to the scope of the node, yielding a set of textual objects for that node.

- Other domain-specific hierarchies are also available, notably in the terminology-rich law domain, which includes the KeySearch taxonomy by WestLaw (`http://west.thomson.com/westlaw/keysearch`) and the Web-based FindLaw hierarchy (`http://www.findlaw.com`) (both of them cross-linked with material relevant for each node).

- The US Patent Classification (`http://www.uspto.gov/go/classification`) and the International Patent Classification (`http://www.wipo.int/classifications/ipc/en`) are exceptionally elaborate taxonomies, where each node is linked to relevant patents.

- The online Wikipedia encyclopedia (`http://www.wikipedia.org`) has a fairly shallow hierarchy but its nodes contain very high-quality articles, which are mostly noise-free (except for occasional spamming).

- In the brick-and-mortar world, library classification systems such as the Universal Decimal Classification (UDC) (Mcilwaine, 2000), the Dewey Decimal Classification (Dewey et al., 2003) or the Library of Congress Classification (Chan, 1999) provide hierarchical structuring of human knowledge for classifying books. By the very virtue of their definition, each hierarchy node can be associated with the text of books cataloged under the node.

In this work we use the ODP as our knowledge base, due to the easy accessibility of its structure and linked resources (cataloged Web sites). However, our methodology is general enough to facilitate other knowledge repositories such as those listed above, and in our future work we intend to explore their utility as well, focusing in particular on the MeSH hierarchy for domain-specific feature generation. In a recent study (Gabrilovich and Markovitch, 2006), we used the Wikipedia encyclopedia as a source of knowledge for feature generation.

A note on terminology is in order here. The most commonly used term for nodes of hierarchical directories of knowledge is "category." In text categorization, however, this term normally refers to topical labels assigned to documents. To prevent possible confusion, we use the word "concept" to refer to the former notion. We represent such concepts as vectors in a high-dimensional space of "attributes." Again, we avoid using the term "features," which is reserved for denoting individual entries of document vectors in text categorization per se.

3.3 Building a Feature Generator

The first step in our methodology is preprocessing, performed once for all future text categorization tasks. We induce a hierarchical text classifier that maps pieces of text onto relevant knowledge concepts, which later serve as generated features. The resulting classifier is called a *feature generator* according to its true purpose in our scheme, as opposed to the text categorizer (or classifier) that we build ultimately. The feature generator represents concepts as vectors of their most characteristic words, which we call *attributes* (reserving the term *features* to denote the properties of documents in text categorization).

The feature generator operates similarly to a regular text classifier—it first learns a classification model in the space of concept attributes, and then identifies a set of concepts that are most appropriate to describe the contents of the input document. Observe that the number of concepts to which the feature generator classifies document text is huge, as suitable knowledge repositories may contain tens and even hundreds of thousands of concepts. Few machine learning algorithms can efficiently handle so many different classes and about an order of magnitude more of training examples. Suitable candidates include the nearest neighbor and the Naive Bayes classifier (Duda and Hart, 1973), as well as prototype formation methods such as Rocchio (Rocchio, 1971) or centroid-based (Han and Karypis, 2000) classifiers. A radically different approach would avoid considering all existing concepts simultaneously, rather, it would work top-down into the hierarchy, identifying several most suitable concepts at each level, as in the hierarchical text classifiers described in the literature (Koller and Sahami, 1997; Dumais and Chen, 2000; Ruiz and Srinivasan, 2002).

3.3.1 Attribute Selection

Prior to learning a text classifier that will act as a feature generator, we represent each concept as an attribute vector. To this end, we pool together all the textual objects for the concept and all of its descendants, and represent the accumulated description with a vector of words. Using all encountered words as attributes is impractical because it yields a classification model that is too big, and because this would inevitably increase the level of noise. The former consideration is essential to allow fitting the induced model into computer memory. The latter consideration is particularly important for Web-based knowledge repositories, which are inherently plagued with noise ranging from intentional directory spamming to merely irrelevant information. To remedy the situation, we perform *attribute selection* for each concept prior to learning the feature generator.

To this end, we use standard attribute selection techniques (Sebastiani, 2002) such as information gain, and identify words that are most characteristic of a concept versus all other concepts. This approach to attribute selection is reminiscent of the approaches described by Chakrabarti et al. (1997) and by Koller and Sahami (1997). Let us denote by D_i the collection of textual objects of c_i and its descendants, $D_i = \{t_{j,k} | c_i \sqsubseteq c_j\}$, and by $\overline{D_i}$ the collection of textual objects for all other concepts, $\overline{D_i} = \{t_{l,k} | c_i \not\sqsubseteq c_l\}$. Then, we can assess the discriminative capacity of each word $w \in D_i$ with respect to $\overline{D_i}$. It is essential to note that conventional attribute selection techniques select attributes for c_i from the entire lexicon, $L = D_i \cup \overline{D_i}$. In our case, however, we aim at selecting words that are most characteristic for the concept, and therefore we limit the selection only to words that actually appear in the textual objects for that concept, i.e., D_i.

Figure 1 shows the algorithm for building a feature generator. The algorithm uses a global structure $Text(c_i)$ that accumulates textual objects for concept c_i and all of its descendants (attributes for the category are then selected from the words occurring in this pool). We manipulate $Text(c_i)$ as an unordered bag of words. Attribute vectors for each category are stored in $Vector(c_i)$.

3.4 Contextual Feature Generation

Feature generation precedes text categorization, that is, before the induction algorithm is invoked to build the text categorizer, the documents are fed to the feature generator.

Traditionally, feature generation uses the basic features supplied with the training instances to construct more sophisticated features. In the case of text processing, however, important information about word ordering will be lost if the traditional approach is applied to the bag of words. Therefore, we argue that feature generation becomes much more powerful when it operates on the raw document text. But should the generator always analyze the whole document as a single unit, as do regular text classifiers?

3.4.1 Analyzing Local Contexts

We believe that considering the document as a single unit can often be misleading: its text might be too diverse to be readily mapped to the right set of concepts, while notions mentioned only briefly may be overlooked. Instead, we propose to partition the document into a series of non-overlapping segments (called *contexts*), and then generate features at this finer level. Each context is classified into a number of concepts in the knowledge base, and pooling these concepts together to describe the entire document results in *multi-faceted* classification. This way, the resulting set of concepts represents the various aspects or sub-topics covered by the document.

Potential candidates for such contexts are simple sequences of words, or more linguistically motivated chunks such as sentences or paragraphs. The optimal resolution for document segmentation can be determined automatically using a validation set. We propose a more principled *multi-resolution* approach that simultaneously partitions the document at several levels of linguistic abstraction (windows of words, sentences, paragraphs, up to taking the entire document as one big chunk), and performs feature generation at each

```
Algorithm BuildFeatureGenerator
  # Compute attribute vectors for all concepts
  BuildVectors($c_0$)

  # Use an induction algorithm to train a feature generator FG
  # using the attribute vectors $Vector(c_i)$
  $FG \leftarrow InduceClassifier(\{Vector(c_i)\})$
  # For feature generation efficiency, build an inverted index
  $InvIndex : w \mapsto \{c_i\}$, s.t. $w \in Vector(c_i)$
----------------------------------------------------------------
Algorithm BuildVectors($c_i$)
  $Text(c_i) = \emptyset$

  # Traverse the hierarchy bottom-up, collecting the textual objects
  # of the descendants of each category
  For each $child \in Children(c_i)$ do
    BuildVectors($child$)
    $Text(c_i) \leftarrow Text(c_i) \cup Text(child)$

  # Now add the textual objects for the category itself
  # along with the optional description (if available)
  $Text(c_i) \leftarrow Text(c_i) \cup \{t_{i,1}, \ldots, t_{i,m}\} \cup Description(c_i)$

  # Build the attribute vector by performing attribute selection
  # among the words of $Text(c_i)$
  $Vector(c_i) \leftarrow AttributeSelection(Text(c_i))$
  # Assign values to the selected attributes
  $Vector(c_i) \leftarrow tfidf(Vector(c_i))$
```

Figure 1: Building a feature generator.

of these levels. We rely on the subsequent *feature selection* step (Section 3.4.2) to eliminate extraneous features, preserving only those that genuinely characterize the document. Figure 2 presents the feature generation algorithm.

In fact, the proposed approach tackles two important problems in natural language processing, namely, *synonymy* (the ability of natural languages to express many notions in more than one way), and *polysemy* (the property of natural language words to convey more than a single sense, while certain words may have as many as dozens of different, sometimes unrelated senses). When individual contexts are classified, *word sense disambiguation* is implicitly performed, thus resolving word polysemy to some degree. A context that contains one or more polysemous words is mapped to the concepts that correspond to the sense *shared* by the context words. Thus, the correct sense of each word is determined with the help of its neighbors. At the same time, enriching document representation with high-level concepts

```
Algorithm FeatureGeneration($D$)
  Let $CT$ be a series of contexts for $D$
  $CT \leftarrow words(D) \cup sentences(D) \cup paragraphs(D) \cup \{D\}$
  Let $F$ be a set of features generated for $D$
  $F \leftarrow \emptyset$
  For each context $ct \in CT$ perform feature generation:
    $F \leftarrow F \cup FG(ct)$
  Represent $D$ as $BagOfWords(D) \cup F$
```

Figure 2: Performing feature generation for document D

and their generalizations addresses the problem of synonymy, as the enhanced representation can easily recognize that two (or more) documents actually talk about related issues, albeit using different vocabularies.

For each context, the feature generator yields a list of concepts ordered by their score, which quantifies their appropriateness to the context. A number of top-scoring concepts are used to actually generate features. For each of these concepts we generate one feature that represents the concept itself, as well an additional group of features that represent ancestors of this concept in the hierarchy of the knowledge repository.

3.4.2 Feature Selection

Using support vector machines in conjunction with bag of words, Joachims (1998) found that SVMs are very robust even in the presence of numerous features. He further observed that the multitude of features are indeed useful for text categorization. These findings were corroborated in more recent studies (Rogati and Yang, 2002; Brank et al., 2002; Bekkerman, 2003) that observed either no improvement or even small degradation of SVM performance after feature selection.[1] Consequently, many later works using SVMs did not apply feature selection at all (Leopold and Kindermann, 2002; Lewis et al., 2004).

This situation changes drastically as we augment the bag of words with generated features. First, nearly any technique for automatic feature generation can easily generate huge numbers of features, which will likely aggravate the "curse of dimensionality." Furthermore, it is feature selection that allows the feature generator to be less than a perfect classifier. When some of the concepts assigned to the document are correct, feature selection can identify them and seamlessly eliminate the spurious ones. We further analyze the utility of feature selection in Section 5.7.

Note also that the categories to which the documents are categorized most likely correspond to a mix of knowledge repository concepts rather than a single one. Therefore, as the feature generator maps documents to a large set of related concepts, it is up to feature selection to retain only those that are relevant to the particular categorization task in hand.

[1] Gabrilovich and Markovitch (2004) described a class of problems where feature selection from the bag of words actually improves SVM performance.

3.4.3 Feature Valuation

In regular text categorization, each word occurrence in document text is initially counted as a unit, and then feature valuation is performed, usually by subjecting these counts to TF.IDF weighting (Salton and Buckley, 1988; Debole and Sebastiani, 2003). To augment the bag of words with generated features and to use a single unified feature set, we need to assign weights to generated features in a compatible manner.

Each generated feature is assigned the basic weight of 1, as in the single occurrence of a word in the bag of words. However, this weight is further multiplied by the classification score produced for each classified concept by the feature generator. This score quantifies the degree of affinity between the concept and the context it was assigned to.

3.4.4 Revisiting the Running Example

Let us revisit the example from Section 1, where we considered a document that belongs to the "copper" category of Reuters-21578. Figure 3 illustrates the process of feature generation for this example. While building the feature generator at the preprocessing stage, our system crawls the Web sites cataloged under mining-related ODP concepts such as Business/Mining_and_Drilling, Science/Technology/Mining and Business/Industrial_Goods_and_Services/Materials/Metals. These include `http://www.teckcominco.com` and `http://www.miningsurplus.com`, which belong to the (now merged) Teck Cominco company. The company's prominence gives it frequent mention in the Web sites we have crawled, and consequently the words "Teck" and "Cominco" are included in the set of attributes selected to represent the above concepts.

During feature generation, the document is segmented into a sequence of contexts The feature generator analyzes these contexts and uses their words (e.g., "Teck" and "Cominco") to map the document to a number of mining-related concepts in the ODP (e.g., Business/Mining_and_Drilling). These concepts, as well as their ancestors in the hierarchy, give rise to a set of generated features that augment the bag of words. Observe that the training documents for the category "copper" underwent similar processing when a text classifier was induced. Consequently, features based on these concepts *were selected* during feature selection and retained in document vectors, thanks to their high predictive capacity. It is due to these features that the document is now categorized correctly, while without feature generation it consistently caused BOW classifiers to err.

4. Using the Open Directory for Feature Generation

We now instantiate the general methodology presented in Section 3 to use the Open Directory project as a knowledge repository.

The Open Directory comprises a hierarchy of approximately 600,000 nodes that catalog over 4,000,000 Web sites, each represented by a URL, a title, and a brief summary of its contents. The directory is organized as a tree where each node has a title (defined by its location within the directory, e.g., Computers/Artificial_Intelligence), and about one-third of all nodes have a short textual description. Every ODP node is associated with a collection of URLs to Web sites cataloged under that node, while each URL has a title and a concise summary of the corresponding Web site. The project constitutes an ongoing effort

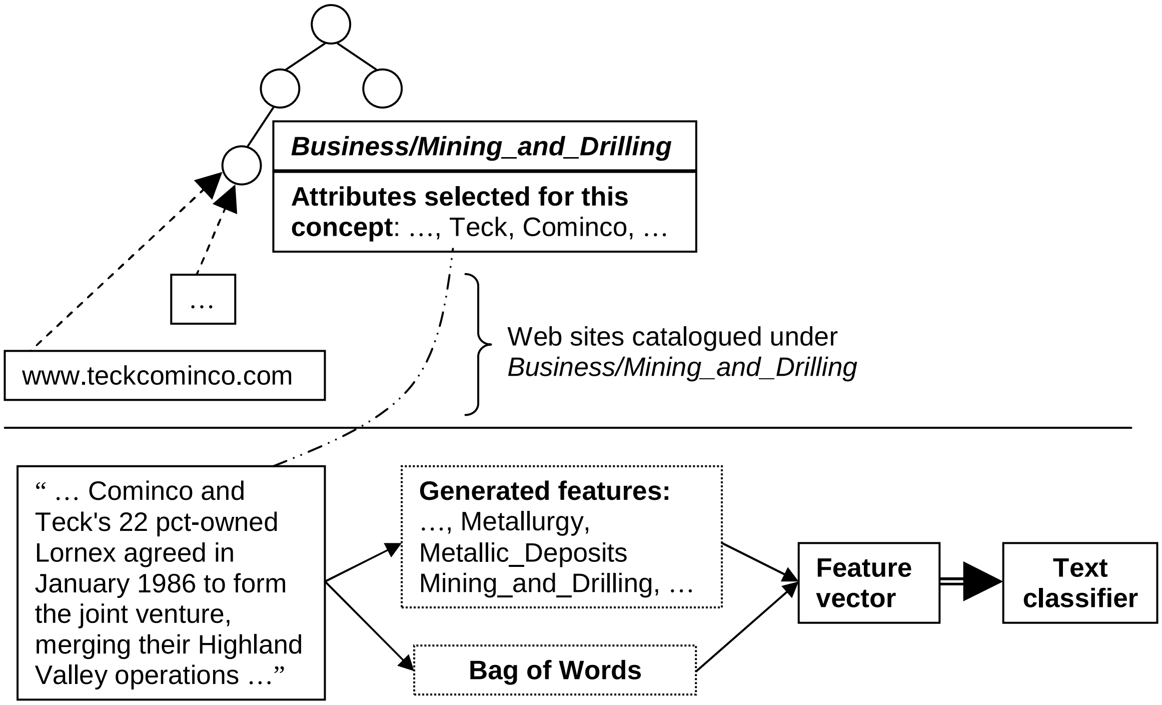
Business/Mining_and_Drilling
Attributes selected for this concept: …, Teck, Cominco, …
…
www.teckcominco.com
Web sites catalogued under Business/Mining_and_Drilling
" … Cominco and Teck's 22 pct-owned Lornex agreed in January 1986 to form the joint venture, merging their Highland Valley operations …"
Generated features: …, Metallurgy, Metallic_Deposits Mining_and_Drilling, …
Bag of Words
Feature vector
Text classifier

Figure 3: Feature generation example

promoted by over 65,000 volunteer editors around the globe, and is arguably the largest publicly available Web directory.[2] Being the result of *pro bono* work, the Open Directory has its share of drawbacks, such as non-uniform coverage, duplicate subtrees in different branches of the hierarchy, and sometimes biased coverage due to peculiar views of the editors in charge. At the same time, however, ODP embeds a colossal amount of human knowledge in a wide variety of areas, covering even very specific scientific and technical concepts.

4.1 Multiplying Knowledge Through Web Crawling

We use the textual descriptions of ODP nodes and their URLs as training examples for learning the feature generator. Although these descriptions alone constitute a sizeable amount of information, we devised a way to increase the volume of training data by several orders of magnitude. We do so by crawling the Web sites pointed at by all cataloged URLs, and obtain a small representative sample of each site. Following the scheme introduced by Yang et al. (2002), each link cataloged in the ODP is used to obtain a small representative sample of the target Web site. To this end, we crawl each cataloged site in the BFS order, starting from the URL listed in the directory. A predefined number of Web pages are downloaded, and then concatenated into a synthetic *meta-document*. This meta-document,

[2] Although the actual size of Yahoo! has not been publicly released in the recent years, it is estimated to be about half the size of the Open Directory. This estimate is based on brute-force exhaustive crawling of the Yahoo! hierarchy. See `http://sewatch.com/reports/directories.html` and `http://www.geniac.net/odp` for more details.

along with the site description listed in the directory, constitutes the textual object for that site. Pooling together the meta-documents for all sites associated with an ODP node gives us a wealth of additional information about it, which we use to enrich the node summary.

4.2 Noise Reduction and Attribute Selection

Using so much knowledge requires a host of filtering mechanisms that control the quality and utility of the generated features. We now describe these mechanisms in detail. In what follows, we distinguish between *structural noise*, which is inherent to the ODP structure, and *content noise*, which is found in the texts we obtain through crawling the cataloged URLs.

4.2.1 Structural noise

However elaborate our knowledge repositories are, they necessarily contain concepts that are detrimental to feature generation. These include concepts too specific or situated too deep in the hierarchy, or having too few textual objects to build a representative attribute vector. It is important to observe, however, that whenever we pruned small categories, we assigned all their textual content to their parents. Here again we benefit from the hierarchical organization of the directory, which allows us to aggregate small fragments of specific knowledge at a higher conceptual level, where its accumulated mass becomes sufficient to define a more general concept.

We identified the following potential sources of noise in the Open Directory:

1. The branch Top/World concentrates material in languages other than English. This entire branch is therefore pruned.

2. Some top-level branches contain concepts that are hardly useful for subsequent text categorization.

   (a) Top/News is a very elaborate subtree devoted to listing numerous CNN stories on various topics organized by date. The nodes of this subtree represent past dates, and do not correspond to useful knowledge concepts.

   (b) Top/Adult lists adult-oriented Web sites, and we believe that the concepts of this subtree are of little use for general purpose text categorization.

   (c) Top/Kids_and_Teens roughly duplicates the structure of the ODP but only lists resources suitable for children.

   All these branches are pruned as well.

3. Overly small categories (usually situated very deep in the hierarchy) that only contain a handful of URLs, and therefore their scope cannot be learned reliably. We therefore eliminate categories with fewer than 10 URLs or those situated below depth level 7 (the textual content of pruned categories is assigned to their parents).

4. The Top/Regional branch contains approximately one third of the entire mass of the ODP data, and is devoted to listing English language sites about various geographical regions of the world. This branch is further divided into continents, countries and

smaller localities, up to the level of cities, towns and landmarks. However, the hierarchy does not stop at this level, and for most localities it provides much more elaborate classification, similar to that of the higher ODP levels. For example, under the path Top/Regional/North_America/United_States/New_York/Localities/N/New_York_City one finds further subdivisions such as Arts_and_Entertainment, Business_and_Economy, Health, Shopping and Society_and_Culture. A similar set of categories duplicating higher-level distinctions (Top/Arts, Top/Business etc.) can be also found in the middle of this path at Top/Regional/North_America/United_States/New_York.

ODP classification principles[3] prescribe that businesses that operate in a particular locality (in this example, local to the State of New York or to New York City) should normally be catalogued under the most specific applicable categories, while businesses with global reach should be catalogued somewhere under Top/Business; the rationale for choosing other categories (e.g., Top/Society/... vs. Top/Regional/North_America/United_States/New_York/Society_and_Culture is similar. However, we believe that when the ODP is used as a knowledge repository to support text categorization, such fine-grained distinctions (e.g., architect offices in Manhattan) are of little use. These categories only pollute the hierarchy with numerous small nodes, each of which only has a small chance of being assigned to any given context.

Therefore, we eliminate overly specific categories under Top/Regional by pruning all paths at the level of geographical names. When the feature generator operates on a context describing a particular New York business, it will map the latter to the New York City node, as well as to one or more appropriate nodes under Top/Business.

5. Web spam, which comes in the form of URLs that are hardly authoritative or representative of their host category, but are nonetheless included in the directory by a minority of unscrupulous editors. We do not explicitly address the problem of spam here, as it lies beyond the scope of our current study.

4.2.2 Content noise

Texts harvested from the WWW are quite different from clean passages in formal written English, and without adequate noise reduction crawled data may do more harm than good. To reduce content noise we perform attribute selection as explained in Section 3.3.1. For example, Table 1 shows the top 10 attributes selected for sample ODP concepts using information gain as the attribute selection criterion. As we can see, the attributes selected for all the sample concepts are very intuitive and plausible.

[3] See `http://dmoz.org/guidelines` and `http://dmoz.org/erz/index.html` for general ODP editorial guidelines, and `http://dmoz.org/Regional/faq.html` for Regional-specific issues.

[4] Many crawled Web pages under Top/Regional/Europe/Switzerland contain non-English material, hence words like “Schweiz” (German for Switzerland) and “der” (German masculine definite article), which survived stop words removal that is only performed for English.

ODP concept	Top 10 selected attributes
Top/Business/Financial_Services	finance, loan, mortgage, equity, insurance, lender, bank, investment, transaction, payment
Top/Computers/Artificial_Intelligence	neural, artificial, algorithm, intelligence, AAAI, Bayesian, probability, IEEE, cognitive, inference
Top/Health/Nutrition	nutrition, diet, nutrient, vitamin, dietary, cholesterol, carbohydrate, intake, protein, fat
Top/Home/Cooking	recipe, sauce, ingredient, soup, salad, casserole, stew, bake, butter, cook
Top/Recreation/Travel	travel, itinerary, trip, destination, cruise, hotel, tour, adventure, travelogue, departure
Top/Regional/Europe/Switzerland[4]	Switzerland, Swiss, Schweiz, und, Suiss, sie, CHF, der, Zurich, Geneva
Top/Science	science, research, scientific, biology, laboratory, analysis, university, theory, study, scientist
Top/Shopping/Gifts	gift, birthday, occasion, basket, card, shipping, baby, keepsake, order, wedding
Top/Society/History	war, history, military, army, civil, historian, soldier, troop, politics, century
Top/Sports/Golf	golf, golfer, tee, hole, fairway, tournament, championship, clubhouse, PGA, par

Table 1: Examples of attribute selection using information gain

4.2.3 Learning the Feature Generator

In our current implementation, the feature generator works as a centroid-based classifier (Han and Karypis, 2000), which represents each category as a centroid vector of the pool of textual objects associated with it.[5] Given a fragment of text supplied as input for feature generation, the classifier represents it as an attribute vector in the same space. It then compares this vector to those of all the concepts, and returns the desired number of best-matching ones. Attribute vectors are compared using the cosine metric (Zobel and Moffat, 1998); the value of the metric is treated as the classification score. A number of top-scoring concepts are retained for each input text as generated features. The feature generator also performs *generalization* of these concepts, and constructs features from the classified concepts *per se* as well as their ancestors in the hierarchy.

5. Empirical Evaluation

To evaluate the utility of knowledge-based feature generation, we implemented the proposed methodology using the Open Directory as a source of world knowledge. Throughout the experiments we used an ODP snapshot as of April 2004. Crawling of URLs cataloged in the Open Directory was performed over the period of April–August 2004.

[5] The centroid classifier offers a simple and efficient way to manage the multitude of concepts in the Open Directory; additional machine learning techniques for learning the feature generator are mentioned in Section 3.3.

5.1 Experimental Methodology

We used the following test collections to evaluate our methodology for feature generation:

1. **Reuters-21578** (Reuters, 1997) is historically the most often used dataset in text categorization research. Following common practice, we used the ModApte split (9603 training, 3299 testing documents) and two category sets, 10 largest categories and 90 categories with at least one training and testing example.

2. **Reuters Corpus Volume I (RCV1)** (Lewis et al., 2004), with over 800,000 documents and three orthogonal category sets, presents a new challenge for text categorization. Since the original RCV1 data contains a number of errors, we used the corrected version RCV1-v2 (Lewis et al., 2004, Section 4). To speed up experimentation, we used a subset of the corpus with 17,808 training documents (dated August 20–27, 1996) and 5341 testing documents (dated August 28–31, 1996). Following the scheme introduced by Brank et al. (2002), we used 16 Topic and 16 Industry categories, which constitute a representative sample of the full groups of 103 and 354 categories, respectively. We also randomly sampled the Topic and Industry categories into 5 sets of 10 categories each. Table 8 (Appendix A) gives the full definition of the category sets we used.

3. **OHSUMED** (Hersh et al., 1994) is a subset of the MEDLINE database, which contains 348,566 references to documents published in medical journals over the period of 1987–1991. Each reference contains the publication title, and about two-thirds (233,445) also contain an abstract. Each document is labeled with several MeSH (MeSH, 2003) categories. There are over 14,000 distinct categories in the collection, with an average of 13 categories per document. Following Joachims (1998), we used a subset of documents from 1991 that have abstracts, taking the first 10,000 documents for training and the next 10,000 for testing. To limit the number of categories for the experiments, we randomly generated 5 sets of 10 categories each. Table 9 (Appendix A) gives the full definition of the category sets we used.

4. **20 Newsgroups (20NG)** (Lang, 1995) is a well-balanced dataset of 20 categories containing 1000 Usenet postings each.

5. **Movie Reviews (Movies)** (Pang et al., 2002) defines a sentiment classification task, where reviews express either positive or negative opinion about the movies. The dataset has 1400 documents in two categories (positive/negative).

We used support vector machines[6] as our learning algorithm to build text categorizers, since prior studies found SVMs to have the best performance for text categorization (Sebastiani, 2002; Dumais et al., 1998; Yang and Liu, 1999). Following established practice, we use the precision-recall break-even point (BEP) to measure text categorization performance. For the two Reuters datasets we report both micro- and macro-averaged BEP, since their categories differ in size significantly. Micro-averaged BEP operates at the document level and is primarily affected by categorization performance on larger categories. On the

[6] We used the SVM^{light} implementation (Joachims, 1999a).

other hand, macro-averaged BEP averages results for individual categories, and thus small categories with few training examples have large impact on the overall performance. For both Reuters datasets we used a fixed data split, and consequently used macro sign test (S-test) (Yang and Liu, 1999) to assess the statistical significance of differences in classifier performance. For 20NG and Movies we performed 4-fold cross-validation, and used paired t-test to assess the significance.

5.2 Implementation Details

In this section we describe the implementation details and design choices of our system.

5.2.1 Constructing the Feature Generator

All ODP data is publicly available in machine-readable RDF format at `http://rdf.dmoz.org`. We used the file `structure.rdf.u8`, which defines the hierarchical structure of the directory, as well as provides category names and descriptions, and the file `content.rdf.u8`, which associates each category with a list of URLs, each having a title and a concise summary of the corresponding Web site. After pruning the Top/World branch, which contains non-English material, and Top/Adult branch, which lists adult-oriented Web sites, we obtained a collection of over 400,000 concepts and 2,800,000 URLs, organized in a very elaborate hierarchy with maximum depth of 13 levels and median depth of 7. Further pruning of too small and deep categories, as well as pruning of the Top/Regional subtree at the level of geographical names as explained in Section 4.2, reduced the number of concepts to 63,000 (the number of URLs was not reduced, since the entire URL population from pruned nodes is moved to their parents).

Textual descriptions of the concepts and URLs amounted to 436 Mb of text (68 Mb in concept titles and descriptions, 368 Mb in URL titles and summaries). In order to increase available information for training the feature generator, we further populated the ODP hierarchy by crawling all of its URLs, and taking the first 10 pages (in the BFS order) encountered at each site to create a representative meta-document of that site. As an additional noise removal step, we discarded meta-documents containing fewer than 5 distinct terms. This operation yielded 425 Gb worth of HTML files. After eliminating all the markup and truncating overly long files at 50 Kb, we ended up with 70 Gb of additional textual data. Compared to the original 436 Mb of text supplied with the hierarchy, we obtained over a 150-fold increase in the amount of data.

Applying our methodology to a knowledge repository of this scale required an enormous engineering effort. After tokenization and removal of stop words, numbers and mixed alphanumeric strings (e.g., “Win2k” or “4Sale”), we obtained 20,800,000 distinct terms. Further elimination of rare words (occurring in less than 5 documents) and applying the Porter stemming algorithm (Porter, 1980) resulted in a more manageable number of 2,900,000 distinct terms that were used to represent ODP nodes as attribute vectors. Up to 1000 most informative attributes were selected for each ODP node using the Document Frequency criterion (other commonly used feature selection techniques, such as Information Gain, χ^2 and Odds Ratio (Yang and Pedersen, 1997; Rogati and Yang, 2002; Mladenic, 1998a), yielded slightly inferior results in text categorization).

In order to speed up consequent classification of document contexts, we also built an *inverted index* that, given a word, provides a list of concepts that have it in their attribute vector (i.e., the word has been *selected* for this concept).

When assigning weights to individual entries in attribute vectors, we took into consideration the location of original word occurrences. For example, words that occurred in URL titles were assigned higher weight than those in the descriptions. Words originating from the descriptions or meta-documents corresponding to links prioritized[7] by the ODP editors were also assigned additional weight. We completely ignored node descriptions since these are only available for about 40% of the nodes, and even then the descriptions are rarely used to actually describe the corresponding concept; in many cases they just contain instructions to the editors or explain what kinds of sites should *not* be classified under the node.

The set of attribute vectors underwent TF.IDF weighting, and eventually served to build a centroid-based feature generator.

5.2.2 Using the Feature Generator

We used the *multi-resolution* approach to feature generation, classifying document contexts at the level of individual words, complete sentences, paragraphs, and finally the entire document.[8] For each context, features were generated from the 10 best-matching ODP concepts produced by the feature generator, as well as for all of their ancestors.

5.2.3 Text Categorization

We conducted the experiments using a text categorization platform of our own design and development named $\mathcal{H}$ogwarts[9] (Davidov et al., 2004). We opted to build a comprehensive new infrastructure for text categorization, as surprisingly few software tools are publicly available for researchers, while those that are available allow only limited control over their operation. $\mathcal{H}$ogwarts facilitates full-cycle text categorization including text preprocessing, feature extraction, construction, selection and valuation, followed by actual classification with cross-validation of experiments. The system currently provides part-of-speech tagging (Brill, 1995), sentence boundary detection, stemming (Porter, 1980), WordNet (Fellbaum, 1998) lookup, a variety of feature selection algorithms, and TF.IDF feature weighting schemes. $\mathcal{H}$ogwarts has over 150 configurable parameters that control its *modus operandi* in minute detail. $\mathcal{H}$ogwarts interfaces with SVM, KNN and C4.5 text categorization algorithms, and computes all standard measures of categorization performance. $\mathcal{H}$ogwarts was designed with a particular emphasis on processing efficiency, and portably implemented in the ANSI C++ programming language. The system has built-in loaders for Reuters-21578 (Reuters, 1997), RCV1 (Lewis et al., 2004), 20 Newsgroups

[7] ODP editors can highlight especially prominent and important Web sites; sites marked as such appear at the top of category listings and are emphasized with an asterisk (in RDF data files, the corresponding links are marked up with a `<priority>` tag).

[8] The 20NG dataset is an exception, owing to its high level of intrinsic noise that renders identification of sentence boundaries extremely unreliable, and causes word-level feature generation to produce too many spurious classifications. Consequently, for this dataset we restrict the multi-resolution approach to individual paragraphs and the entire document only.

[9] *Hogwarts School of Witchcraft and Wizardry* is the educational institution attended by Harry Potter (Rowling, 1997).

(Lang, 1995), Movie Reviews (Pang et al., 2002), and OHSUMED (Hersh et al., 1994), while additional datasets can be easily integrated in a modular way.

In the preprocessing step, each document undergoes the following. Document text is first tokenized, and title words are replicated twice to emphasize their importance. Then, stop words, numbers and mixed alphanumeric strings are removed, and the remaining words are stemmed. The bag of words is next merged with the set of features generated for the document by analyzing its contexts as explained in Section 3.4, and rare features occurring in fewer than 3 documents are removed.

Since earlier studies found that most BOW features are indeed useful for SVM text categorization (Section 3.4.2), we take the bag of words in its entirety (with the exception of rare features removed in the previous step). The generated features, however, undergo feature selection using the information gain criterion. Finally, feature valuation is performed using the "ltc" TF.IDF function (logarithmic term frequency and inverse document frequency, followed by cosine normalization) (Salton and Buckley, 1988; Debole and Sebastiani, 2003).

5.3 Qualitative Analysis of Feature Generation

We now study the process of feature generation on a number of actual examples.

5.3.1 Feature Generation per se

In this section we demonstrate ODP-based feature generation for a number of sample sentences taken from CNN and other Web sites. For each example, we discuss a number of highly relevant features found among the top ten generated ones. Online Appendix A (`http://www.cs.technion.ac.il/~gabr/jmlr2006-online-appendix.html`) gives all 10 classifications produced for each context (some of these classifications are less relevant, and are consequently removed during feature selection, as explained in Section 3.4.2 and illustrated in Section 5.3.3.

- **Text:** *"Rumsfeld appeared with Gen. Richard Myers, chairman of the Joint Chiefs of Staff."*

  Sample generated features:

  - Society/Issues/Government_Operations, Society/Politics — both Donald Rumsfeld and Richard Myers are senior government officers, hence the connection to government operations and politics. Their names have been selected for these ODP concepts, since they appear in many Web sites cataloged under them, such as the National Security Archive at the George Washington University (`http://www.gwu.edu/~nsarchiv`)and the John F. Kennedy School of Government at Harvard University (`http://www.ksg.harvard.edu`).
  - Society/Issues/Warfare_and_Conflict/Specific_Conflicts/Iraq, Science/Technology/Military_Science, Society/Issues/Warfare_and_Conflict/Weapons—again, both persons mentioned were prominent during the Iraq campaign.

  - Society/History/By_Region/North_America/United_States/Presidents/Bush,_George_Walker — Donald Rumsfeld serves as Secretary of Defense under President George W. Bush
  - Society/Politics/Conservatism — Rumsfeld is often seen as holding conservative views on a variety of political issues.

- **Text:** *"The new film follows Anakin's descent into evil and lust for power."*

  Sample generated features:

  - Arts/Movies/Titles/Star_Wars_Movies is the root of the ODP subtree devoted to the "Star Wars" movie series. The word "Anakin" has been selected as an attribute for this concept due to its numerous occurrences in the cataloged Web sites such as `http://www.theforce.net` and `http://www.starwars.com`.
  - Arts/Performing_Arts/Acting/Actors_and_Actresses/Christensen,_Hayden is the actor who played Anakin Skywalker; this particular piece of information cannot be inferred from the short input sentence without elaborate background knowledge.

- **Text:** *"On a night when Dirk Nowitzki (34 points), Jerry Stackhouse (29), Josh Howard (19) and Jason Terry (17) all came up big, he couldn't match their offensive contributions."*

  Sample generated features:

  - Sports/Basketball/Professional/NBA/Dallas_Mavericks—even though the sentence mentions neither the particular sport nor the name of the team, the power of context is at its best, immediately yielding the correct classification as the best-scoring generated feature. The names of the players mentioned in the context occur often in the Web sites cataloged under this concept, including such resources as `www.nba.com/mavericks`, `http://dallasbasketball.com` and `sports.yahoo.com/nba/teams/dal`.

- **Text:** *"Herceptin is a so-called targeted therapy because of its ability to attack diseased cells and leave healthy ones alone."*

  Sample generated features:

  - Health/Conditions_and_Diseases/Cancer/Breast, Society/Issues/Health/Conditions_and_Diseases/Cancer/Alternative_Treatments, Health/Support_Groups/Conditions_and_Diseases/Cancer provide relevant additional information for Herceptin, a medication for breast cancer. The name of this medicine has been selected for these concepts due to its occurrences in cataloged Web sites such as `www.breastcancer.org`, `www.hopkinsmedicine.org/breastcenter` and `cancer.gov/cancerinfo/wyntk/breast`.

- Finally, we give an example of how the power of context can be used for word sense disambiguation. The following pair of sentences use the word "tie" in two different meanings—once as a necktie and once as a kind of connection. Even though these sentences contain no distinguishing proper names, the context of the polysemous words allows the feature generator to produce correct suggestions in both cases

Text: *"Kinship with others is based either on blood ties or on marital ties."*

Sample generated features:

- Society/Genealogy
- Home/Family
- Society/Relationships
- Science/Social_Sciences/Sociology

Text: *"Our tie shop includes plain solid colour ties, novelty ties, patterned silk ties, and men's bow ties."*

Sample generated features:

- Shopping/Clothing/Mens/Neckties
- Shopping/Clothing/Accessories/Mens
- Business/Consumer_Goods_and_Services/Clothing/Accessories/Ties_and_Scarves

Evidently, many of the generated features could not have been accessed by conventional text classification methods, since heavy use of world knowledge is required to deduce them.

5.3.2 Actual Text Categorization Examples Under a Magnifying Glass

Thanks to feature generation, our system correctly classifies the running example document #15264. Let us consider additional testing examples from Reuters-21578 that are incorrectly categorized by the BOW classifier. Document #16143 belongs to the category "money-fx" (money/foreign exchange) and discusses the devaluation of the Kenyan shilling. Even though "money-fx" is one of the 10 largest categories, the word "shilling" does not occur in its training documents even once. However, the feature generator easily recognizes it as a kind of currency, and produces features such as Recreation/Collecting/Paper_Money and Recreation/Collecting/Coins/World_Coins. While analyzing document contexts it also uses other words such as "Central Bank of Kenya" and "devaluation" to correctly map the document to ODP concepts Society/Government/Finance, Science/Social_Sciences/Economics and Business/Financial_Services/Banking_Services. Even though the behavior of the Kenyan shilling was never mentioned in the training set, these high-level features were also constructed for many training examples, and consequently the document is now classified correctly.

Similarly, document #18748 discusses Italy's balance of payments and belongs to the category "trade" (interpreted as an economic indicator), while the word "trade" itself does not occur in this short document. However, when the feature generator considers document contexts discussing Italian deficit as reported by the Bank of Italy, it correctly maps them to concepts such as Society/Government/Finance, Society/Issues/Economic/International/Trade, Business/International_Business_and_Trade. These features, which were also generated for training documents in this category (notably, document #271 on Japanese trade surplus, document #312 on South Korea's account surplus, document #354 on tariff cuts in Taiwan and document #718 on U.S.-Canada trade pact), allow the document to be categorized correctly.

Let us also consider a few documents from the Movie Reviews dataset that confuse the BOW classifier (here we consider a training/testing split induced by one particular cross-validation fold). Recall that this dataset represents a sentiment classification task, where documents are classified according to the sentiment of the review (positive or negative) rather than its topic. Document #19488 contains a negative review of Star Wars Episode 1, but at the word level it is difficult to judge its true sentiment since positive and negative words are interspersed. For instance, the sentence "Anakin is annoying and unlikeable, instead of cute and huggable as Lucas no doubt intended" contains two words with positive connotation ("cute and huggable") that counterbalance the two words with negative ones ("annoying and unlikeable"). However, given contexts like "The two leads are hideously boring, static characters given little to do and too much time to do it," the feature generator produces features such as Arts/Movies/Reviews/Top_Lists/Bad_Films. This ODP node catalogs Web sites devoted to reviews of bad movies, and the wording of this sample context looks similar to that used in known negative reviews (as cataloged in the ODP). In fact, this particular feature is one of the most informative ones generated for this dataset, and it is also produced for contexts like "Next up we have the dialogue, which is amusingly bad at its best, painful at its worst" and "What ensues is a badly scripted and horribly directed 114 minutes of cinema hell," both found in negative reviews.

As another example, consider document #15111, which contains a positive review of the movie "Soldier." This review, which constantly switches between criticizing and praising the film, easily perplexes the BOW classifier. Interestingly, given the sentence "It is written by David Webb Peoples, who penned the screenplay to the classic Blade Runner and the critically-acclaimed 12 Monkeys," the feature generator constructs the highly informative feature Arts/Movies/Reviews/Top_Lists/Good_Films. This is made possible by the references to known good films ("Blade Runner" and "12 Monkeys") that are listed in Web sites devoted to good films (`http://www.filmsite.org` and `http://us.imdb.com/top_250_films`, for example). The same feature was also generated for a number of training documents, and thus helps the classifier to categorize the document correctly.

5.3.3 The Importance of Feature Selection

To understand the utility of feature selection, consider a sample sentence from our running example, Reuters document #15264: "Cominco's share of production was 43,000 short tons of copper, 340,000 ounces of silver and 800 ounces of gold." Table 2 gives the top ten ODP concepts generated as features for this context. Most of the assigned concepts deal with mining and drilling, and will eventually be useful features for document classification. However, the concepts Business/Investing/ Commodities,_Futures/Precious_Metals, Shopping and Business/Investing/Commodities,_Futures/ Precious_Metals/Gold have been triggered by the words "gold" and "silver," which are mentioned incidentally and do not describe the gist of the document. Feature selection is therefore needed to eliminate features based on these extraneous concepts.

As another example, consider the following sentence taken from the same document: "'Cominco, 29.5 percent owned by a consortium led by Teck, is optimistic that the talks

#	ODP concept
1	Business/Mining_and_Drilling/Mineral_Exploration_and_Extraction
2	Business/Mining_and_Drilling
3	Business/Mining_and_Drilling/Mineral_Exploration_and_Extraction/Base_Metals
4	Science/Technology/Mining
5	Business/Mining_and_Drilling/Consulting
6	Business/Investing/Commodities,_Futures/Precious_Metals
7	Shopping
8	Business/Mining_and_Drilling/Mining_Equipment
9	Business/Investing/Commodities,_Futures/Precious_Metals/Gold
10	Science/Technology/Mining/Investments

Table 2: The top ten ODP concepts generated for the sentence "Cominco's share of production was 43,000 short tons of copper, 340,000 ounces of silver and 800 ounces of gold."

will soon be concluded,' spokesman Don Townson told Reuters," along with its top ten classifications given in Table 3. Here, the concept Society/Issues is generated by the word "Reuters." In turn, the concept Business/Marketing_and_Advertising/Consulting/Sales is triggered by the name of the company spokesman, Don Townson. As it happens, a sales consulting company named "Townson & Alexander Consulting Services" is catalogued under this concept. Based on the crawled content of this site, the word "Townson" and other sales-related words in the context (e.g., "percent," "owned," "optimistic," and "consortium") taken together yield this concept in the results. Again, this sales-related concept is hardly useful for categorizing copper-related documents, and features based on it would therefore not be selected.

5.4 The Effect of Feature Generation

We first demonstrate that the performance of basic text categorization in our implementation (column "Baseline" in Table 4) is consistent with the state of the art as reflected in other published studies (all using SVM). On Reuters-21578, Dumais et al. (1998) achieved micro-BEP of 0.920 for 10 categories and 0.870 for all categories. On 20NG, Bekkerman (2003) obtained BEP of 0.856. Pang et al. (2002) obtained accuracy of 0.829 on Movies. The minor variations in performance are due to differences in data preprocessing in the different systems; for example, for the Movies dataset we worked with raw HTML files rather than with the official tokenized version, in order to recover sentence and paragraph structure for contextual analysis. For RCV1 and OHSUMED, direct comparison with published results is more difficult because we limited the category sets and the date span of documents to speed up experimentation.

Table 4 shows the results of using feature generation for text categorization, with significant improvements ($p < 0.05$) shown in bold. For both Reuters datasets, we consistently

#	ODP concept
1	Business/Mining_and_Drilling/Mineral_Exploration_and_Extraction/Base_Metals
2	Business/Mining_and_Drilling/Mineral_Exploration_and_Extraction
3	Business/Mining_and_Drilling
4	Business/Mining_and_Drilling/Consulting
5	Society/Issues
6	Regional/North_America/Canada/British_Columbia/Localities/Kimberley
7	Science/Technology/Mining
8	Business/Marketing_and_Advertising/Consulting/Sales
9	Regional/North_America/Canada/Quebec/Regions/Northern_Quebec
10	Science/Environment/Mining

Table 3: The top ten ODP concepts generated for the sentence “‘Cominco, 29.5 percent owned by a consortium led by Teck, is optimistic that the talks will soon be concluded,’ spokesman Don Townson told Reuters.”

observed larger improvements in macro-averaged BEP, which is dominated by categorization effectiveness on small categories. This goes in line with our expectations that the contribution of external knowledge should be especially prominent for categories with few training examples. As can be readily seen, categorization performance was improved for all datasets, with notably high improvements for Reuters RCV1, OHSUMED and Movies. We believe these results clearly demonstrate the advantage of knowledge-based feature generation.

5.5 The Effect of Contextual Analysis

We now explore the various possibilities for defining document contexts for feature generation, i.e., chunks of document text that are classified onto the ODP to construct features. Figure 4 shows how text categorization performance on the Movies dataset changes for various contexts. The x-axis measures context length in words, and the *FG/words* curve corresponds to applying the feature generator to the context of that size. With these word-level contexts, maximum performance is achieved when using pairs of words (x=2). The *Baseline* line represents text categorization without feature generation. The *FG/doc* line shows what happens when the entire document is used as a single context. In this case, the results are somewhat better than without feature generation (*Baseline*), but are still inferior to the more fine-grained word-level contexts (*FG/words*). However, the best performance by far is achieved with the multi-resolution approach (*FG/multi*), in which we use a series of linguistically motivated chunks of text, starting with individual words, and then generating features from sentences, paragraphs, and finally the entire document.

5.6 The Effect of Knowledge Breadth

In the experiments reported in Section 5.4 we performed feature generation using the entire ODP. It is interesting to observe, however, that four out of the five datasets we used have

Dataset	Baseline		Feature generation		Improvement vs. baseline	
	micro BEP	macro BEP	micro BEP	macro BEP	micro BEP	macro BEP
Reuters-21578						
10 categories	0.925	0.874	0.930	0.884	+0.5%	+1.1%
90 categories	0.877	0.602	0.880	0.614	+0.3%	+2.0%
RCV1						
Industry-16	0.642	0.595	0.648	**0.613**	+0.9%	**+3.0%**
Industry-10A	0.421	0.335	**0.457**	**0.420**	**+8.6%**	**+25.4%**
Industry-10B	0.489	0.528	**0.530**	**0.560**	**+8.4%**	**+6.1%**
Industry-10C	0.443	0.414	**0.468**	**0.463**	**+5.6%**	**+11.8%**
Industry-10D	0.587	0.466	0.588	**0.496**	+0.2%	**+6.4%**
Industry-10E	0.648	0.605	0.657	**0.639**	+1.4%	**+5.6%**
Topic-16	0.836	0.591	0.840	**0.660**	+0.5%	**+11.7%**
Topic-10A	0.796	0.587	0.803	**0.692**	+0.9%	**+17.9%**
Topic-10B	0.716	0.618	0.727	**0.655**	+1.5%	**+6.0%**
Topic-10C	0.687	0.604	0.694	**0.618**	+1.0%	**+2.3%**
Topic-10D	0.829	0.673	0.836	0.687	+0.8%	+2.1%
Topic-10E	0.758	0.742	0.762	0.756	+0.5%	+1.9%
OHSUMED						
OHSUMED-10A	0.518	0.417	**0.537**	**0.479**	**+3.7%**	**+14.9%**
OHSUMED-10B	0.656	0.500	0.659	**0.548**	+0.5%	**+9.6%**
OHSUMED-10C	0.539	0.505	0.547	**0.540**	+1.5%	**+6.9%**
OHSUMED-10D	0.683	0.515	0.688	**0.549**	+0.7%	**+6.6%**
OHSUMED-10E	0.442	0.542	0.452	**0.573**	+2.3%	**+5.7%**
20NG	0.854		**0.858**		**+0.5%**	
Movies	0.813		**0.842**		**+3.6%**	

Table 4: Text categorization with and without feature generation

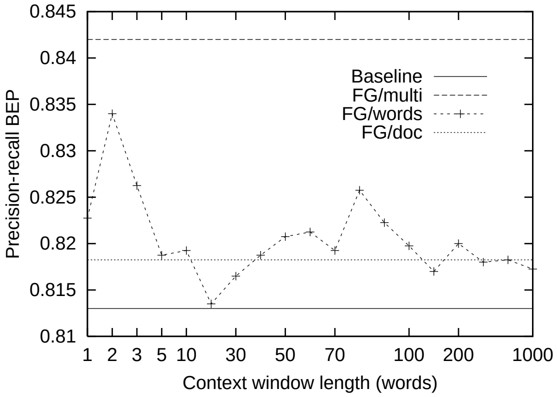

Figure 4: Varying context length (Movies)

Dataset	Domain-specific ODP subset			Full ODP	
	Subset description	micro BEP	macro BEP	micro BEP	macro BEP
Reuters-21578	Top/Business				
10 categories		+0.4%	+0.6%	+0.5%	+1.1%
90 categories		+0.1%	+1.2%	+0.3%	+2.0%
RCV1	Top/Business				
Industry-16		+1.9%	+2.2%	+0.9%	**+3.0%**
Topic-16		+0.5%	+1.4%	+0.5%	**+11.7%**
OHSUMED	Top/Health				
OHSUMED-10A		+2.1%	+1.7%	**+3.7%**	**+14.9%**
OHSUMED-10B		+0.2%	+1.2%	+0.5%	**+9.6%**
OHSUMED-10C		+1.7%	+2.8%	+1.5%	**+6.9%**
OHSUMED-10D		+0.3%	+1.9%	+0.7%	**+6.6%**
OHSUMED-10E		+2.7%	+1.8%	+2.3%	**+5.7%**
OHSUMED	Top/Health + Top/Science				
OHSUMED-10A		+5.4%	+3.6%	**+3.7%**	**+14.9%**
OHSUMED-10B		+0.3%	+3.4%	+0.5%	**+9.6%**
OHSUMED-10C		+0.6%	+3.8%	+1.5%	**+6.9%**
OHSUMED-10D		+0.9%	+5.8%	+0.7%	**+6.6%**
OHSUMED-10E		+1.6%	+1.8%	+2.3%	**+5.7%**
Movies	Top/Arts	**+2.6%**		**+3.6%**	

Table 5: Text categorization with and without feature generation, when only a subset of ODP is used

a fairly narrow scope.[10] Specifically, both Reuters datasets (Reuters-21578 and RCV1) contain predominantly economic news and therefore match the scope of the Top/Business branch of the ODP. Similarly, Movie Reviews contains opinions about movies, and therefore fits the scope of Top/Arts. OHSUMED contains medical documents, which can be modelled within the scope of Top/Health and Top/Science. In light of this, it could be expected that restricting the feature generator to a particular ODP branch that corresponds to the scope of the test collection would result in much better categorization accuracy due to the elimination of noise in "unused" ODP branches.

Experimental results (Table 5) disprove this hypothesis. As can be seen, in the absolute majority of cases the improvement over the baseline is much smaller than when the entire ODP is used (cf. Table 4). These findings show the superiority of wide general-purpose knowledge over its domain-specific subsets.

[10] The 20 Newsgroups dataset consists of 20 diverse categories, each of which corresponds to one or more ODP branches.

5.7 The Utility of Feature Selection

Under the experimental settings defined in Section 5.2, feature generation constructed approximately 4–5 times as many features as are in the bag of words (after rare features that occurred in less than 3 documents were removed). We conducted two experiments to understand the effect of feature selection in conjunction with feature generation.

Since earlier studies found that feature selection from the bag of words impairs SVM performance (Section 3.4.2), we first apply it only to the generated features and use the selected ones to augment the (entire) bag of words. In Figures 5 and 6, the *BOW* line depicts the baseline performance without generated features, while the *BOW+GEN* curve shows the performance of the bag of words augmented with progressively larger fractions of generated features (sorted by information gain). For both datasets, the performance peaks when only a small fraction of the generated features are used, while retaining more generated features has a noticeable detrimental effect.

Our second experiment examined the performance of the generated features alone, without the bag of words (*GEN* curve in Figures 5 and 6). For Movies, discarding the BOW features leads to somewhat worse performance, but the decrease is far less significant than what could be expected—using only the generated features we lose less than 3% in BEP compared with the BOW baseline. For 20NG, a similar experiment sacrifices about 10% of the BOW performance, as this dataset is known to have a very diversified vocabulary, for which many studies found feature selection to be particularly harmful. Similarly, for OHSUMED, using only the generated features sacrifices up to 15% in performance, reinforcing the value of precise medical terminology that is discarded in this experiment. However, the situation is reversed for both Reuters datasets. For Reuters-21578, the generated features alone yield a 0.3% improvement in micro- and macro-BEP for 10 categories, while for 90 categories they only lose 0.3% in micro-BEP and 3.5% in macro-BEP compared with the bag of words. For RCV1/Industry-16, disposing of the bag of words reduces BEP performance by 1–3%. Surprisingly, for RCV1/Topic-16 (Figure 6) the generated features *per se* command a 10.8% improvement in macro-BEP, rivalling the performance of *BOW+GEN*, which gains only another 1% (Table 4). We interpret these findings as further reinforcement that the generated features improve the quality of the representation.

5.8 The Effect of Category Size

We saw in Section 5.4 that feature generation greatly improves text categorization for smaller categories, as can be evidenced in the greater improvements in macro-BEP. To explore this phenomenon further, we depict in Figures 7 and 8 the relation between the category size and the improvement due to feature generation for RCV1 (the number of categories in each bin appears in parentheses above the bars). To this end, we pooled together the categories that comprised the individual sets (10A–10E) in the Industry and Topic groups, respectively.

As we can readily see, smaller categories tend to benefit more from knowledge-based feature generation. These graphs also explain the more substantial improvements observed for Industry categories compared to Topic categories—as can be seen from the graphs, Topic categories are larger than Industry categories, and the average size of Topic categories (among those we used in this study) is almost 6 times larger than that of Industry categories.

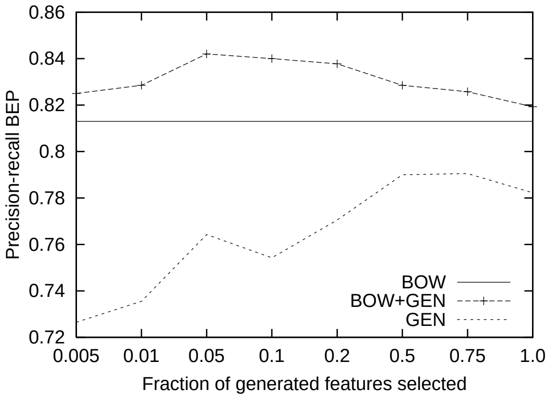

Figure 5: Feature selection (Movies)

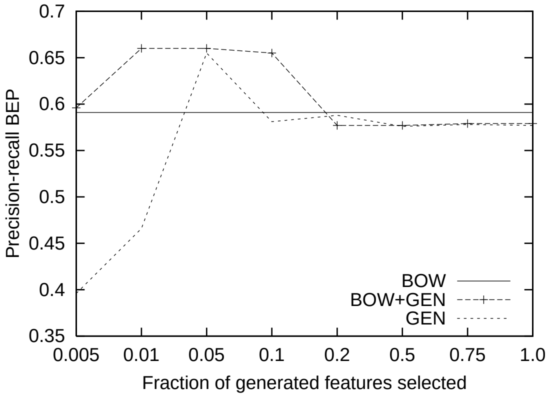

Figure 6: Feature selection (RCV1/Topic-16)

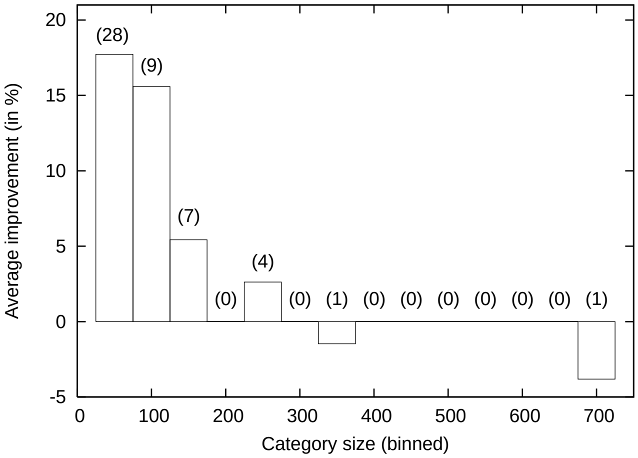

Figure 7: RCV1 (Industry): Average improvement versus category size

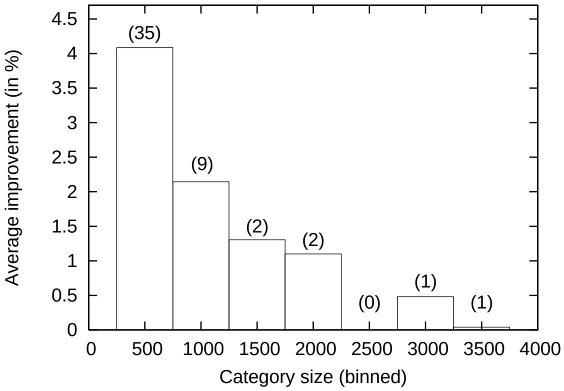

Figure 8: RCV1 (Topic): Average improvement versus category size

5.9 The Effect of Feature Generation for Classifying Short Documents

We conjectured that knowledge-based feature generation might be particularly useful for classifying short documents. To evaluate this hypothesis, we derived several datasets of short documents based on the test collections listed in Section 5.1. Recall that about one-third of the references in OHSUMED have titles but no abstract and can therefore be considered short documents "as-is." We used the same range of documents as in Section 5.1, but considered only those without abstracts. This yielded 4,714 training and 5,404 testing documents. For all other datasets, we created a short document from each original document by taking only the title of the latter (with the exception of Movie Reviews, where documents do not have titles). It should be noted, however, that substituting a title for the full document is a poor man's way to obtain a collection of classified short documents. When documents were first labeled with categories, the human labeller saw the document *in its entirety*. In particular, a category might have been assigned to a document on the basis of some facts mentioned in its body, even though the relevant facts may well be missing from the (short) title. Thus, taking all the categories of the original documents to be "genuine" categories of the title is often misleading. However, because we know of no publicly available test collections of short documents, we decided to use the datasets constructed as explained above. Interestingly, OHSUMED documents without abstracts have been classified as such by humans; working with the OHSUMED-derived dataset can thus be considered a "pure" experiment.

Table 6 presents the results of this experiment. As we can see, in the majority of cases (except for RCV1 Topic category sets), feature generation leads to greater improvement on short documents than on regular documents. Notably, the improvements are particularly high for OHSUMED, where "pure" experimentation on short documents is possible (see above).

5.10 Processing Time

Using the ODP as a source of background knowledge requires additional computation. This extra computation includes the (one-time) preprocessing step where the feature generator

Dataset	Short documents						Full documents	
	Baseline		Feature generation		Improvement vs. baseline		Improvement vs. baseline	
	micro BEP	macro BEP	micro BEP	macro BEP	micro BEP	macro BEP	micro BEP	macro BEP
Reuters-21578								
10 categories	0.868	0.774	0.868	0.777	+0.0%	+0.4%	+0.5%	+1.1%
90 categories	0.793	0.479	0.794	0.498	+0.1%	**+4.0%**	+0.3%	+2.0%
RCV1								
Industry-16	0.454	0.400	0.466	**0.415**	+2.6%	**+3.7%**	+0.9%	**+3.0%**
Industry-10A	0.249	0.199	**0.278**	**0.256**	**+11.6%**	**+28.6%**	**+8.6%**	**+25.4%**
Industry-10B	0.273	0.292	**0.348**	**0.331**	**+27.5%**	**+13.4%**	**+8.4%**	**+6.1%**
Industry-10C	0.209	0.199	**0.295**	**0.308**	**+41.1%**	**+54.8%**	**+5.6%**	**+11.8%**
Industry-10D	0.408	0.361	**0.430**	**0.431**	**+5.4%**	**+19.4%**	+0.2%	**+6.4%**
Industry-10E	0.450	0.410	**0.490**	**0.459**	**+8.9%**	**+12.2%**	+1.4%	**+5.6%**
Topic-16	0.763	0.529	0.763	0.534	+0.0%	+0.9%	+0.5%	**+11.7%**
Topic-10A	0.718	0.507	0.720	0.510	+0.3%	+0.6%	+0.9%	**+17.9%**
Topic-10B	0.647	0.560	0.644	0.560	-0.5%	+0.0%	+1.5%	**+6.0%**
Topic-10C	0.551	0.471	0.561	0.475	+1.8%	+0.8%	+1.0%	**+2.3%**
Topic-10D	0.729	0.535	0.730	**0.553**	+0.1%	**+3.4%**	+0.8%	+2.1%
Topic-10E	0.643	0.636	0.656	0.646	+2.0%	+1.6%	+0.5%	+1.9%
OHSUMED								
OHSUMED-10A	0.302	0.221	**0.357**	**0.253**	**+18.2%**	**+14.5%**	**+3.7%**	**+14.9%**
OHSUMED-10B	0.306	0.187	**0.348**	**0.243**	**+13.7%**	**+29.9%**	+0.5%	**+9.6%**
OHSUMED-10C	0.441	0.296	**0.494**	**0.362**	**+12.0%**	**+22.3%**	+1.5%	**+6.9%**
OHSUMED-10D	0.441	0.356	0.448	**0.419**	+1.6%	**+17.7%**	+0.7%	**+6.6%**
OHSUMED-10E	0.164	0.206	**0.211**	**0.269**	**+28.7%**	**+30.6%**	+2.3%	**+5.7%**
20NG	0.699		**0.740**		**+5.9%**		**+0.5%**	

Table 6: Text categorization of short documents with and without feature generation. (The improvement percentage in the two rightmost columns is computed relative to the baseline shown in Table 4.)

Dataset	Number of documents	Number of words[12]
20NG	19,997	5.5 million
Movies	1,400	0.95 million
Reuters-21578	21,902	2.8 milion
RCV1		
- full	804,414	196 million
- used in this study	23,149	5.5 million
OHSUMED		
- full	348,566	57 million
- used in this study	20,000	3.7 million

Table 7: Test collections sizes

is built, as well as the actual feature generation performed on documents prior to text categorization. The processing times reported below were measured on a workstation with dual Xeon 2.2 GHz CPU and 2 Gb RAM running the Microsoft Windows XP Professional operating system (Service Pack 1).

Parsing the ODP structure (file `structure.rdf.u8`) took 3 minutes. Parsing the list of ODP URLs (file `content.rdf.u8`) required 3 hours, and parsing the crawled ODP data (meta-documents collected from all cataloged URLs) required 2.6 days. Attribute selection for ODP concepts took 1.5 hours. The cumulative one-time expenditure for building the feature generator was therefore just under 3 days (not counting the actual Web crawling that was performed beforehand).

We benchmarked feature generation in two scenarios—individual words and 10-word windows. In the former case, the feature generator classified approximately 310 words per second, while in the latter case it classified approximately 45 10-word windows per second (i.e., 450 words per second).[11] These times constitute the additional overhead required by feature generation compared with regular text categorization. Table 7 lists the sizes of the test collections we experimented with (see Section 5.1). To speed up experimentation, we used subsets of the entire RCV1 and OHSUMED collections; these subsets comparable in size with 20 Newsgroups and Reuters-21578.

In the light of the improvements in categorization accuracy that we report in Section 5.4, we believe that the extra processing time is well compensated for. In operational text categorization systems, documents rarely arrive in huge batches of hundreds of thousands at a time. For example, the RCV1 dataset contains all English-language news items published by Reuters over the period of one year. Therefore, in practical settings, once the classification model has been trained, the number of documents it needs to classify per time unit is much more reasonable, and can be easily facilitated by our system.

[11] Classifying word windows is more efficient due to the sharing of data structures when processing the words in a single context.

[12] Measured using the ‘`wc`’ utility available on UNIX systems.

6. Related Work

To date, quite a few attempts have been made to deviate from the orthodox bag of words paradigm, usually with limited success. In particular, representations based on phrases (Lewis, 1992; Dumais et al., 1998; Fuernkranz et al., 1998), named entities (Kumaran and Allan, 2004), and term clustering (Lewis and Croft, 1990; Bekkerman, 2003) have been explored. However, none of these techniques could possibly overcome the problem underlying the various examples we reviewed in this paper—lack of world knowledge.

In mainstream information retrieval, query expansion techniques are used to augment queries with additional terms. However, this approach does not enhance queries with high-level concepts beyond words or phrases (as this would require indexing the entire document collection accordingly). It occasionally uses WordNet (Fellbaum, 1998) as a source of external knowledge, but queries are more often enriched with individual words, which are chosen either through relevance feedback (Mitra et al., 1998; Xu and Croft, 2000), or by consulting dictionaries and thesauri (Voorhees, 1994, 1998). Ballesteros and Croft (1997) studied query expansion with phrases in the context of cross-lingual information retrieval.

Feature generation techniques were found useful in a variety of machine learning tasks (Markovitch and Rosenstein, 2002; Fawcett, 1993; Matheus, 1991). These techniques search for new features that describe the target concept better than the ones supplied with the training instances. A number of proposed feature generation algorithms (Pagallo and Haussler, 1990; Matheus and Rendell, 1989; Hu and Kibler, 1996; Murphy and Pazzani, 1991) led to significant improvements in performance over a range of classification tasks. However, even though feature generation is an established research area in machine learning, only a few works have applied it to text processing (Kudenko and Hirsh, 1998; Mikheev, 1999; Cohen, 2000). It is interesting to observe that traditional machine learning data sets, such as those available from the UCI data repository (Blake and Merz, 1998), are only available as feature vectors, while their feature set is essentially fixed. Textual data, however, is almost always available in raw text format. Thus, in principle, possibilities for feature generation are more plentiful and flexible.

Kudenko and Hirsh (1998) proposed a domain-independent feature generation algorithm that uses Boolean features to test whether certain sub-sequences appear a minimum number of times. They applied the algorithm to three toy problems in topic spotting and book passage categorization. Mikheev (1999) used a feature collocation lattice as a feature generation engine within a maximum entropy framework and applied it to document categorization, sentence boundary detection, and part-of-speech tagging. This work utilized information about individual words, bigrams and trigrams to prebuild the feature space. A set of feature cliques with the highest log-likelihood estimate was then selected. Cohen (2000) researched the problem of automatically discovering features useful for classification according to the given labels, given a set of labeled instances *not accompanied by a feature set*. Problems of this kind occur, for example, when classifying names of musical artists by music genres, or names of computer games by categories such as quest or action. The paper proposed to collect relevant Web pages, and then define features based on words from HTML headers that co-occur with the names to be classified. The fact that a word appears in an HTML header usually signifies its importance, and hence potential usefulness, for classification. The author also identified another source of features on the basis of their *positions* inside HTML documents, where position is defined as a sequence of tags in the HTML parsing tree, between the root of the tree and the name of interest. For example, if a name appears frequently in tables, this characteristic may be defined as a feature.

Several studies performed feature construction using WordNet and other domain-specific dictionaries (Scott, 1998; Urena-Lopez et al., 2001; Bloehdorn and Hotho, 2004; Wang et al., 2003). Scott (1998) completely replaced a bag of words with a bag of synsets[13]. Urena-Lopez et al. (2001) used WordNet in conjunction with Rocchio (Rocchio, 1971) and Widrow-Hoff (Lewis et al., 1996; Widrow and Stearns, 1985, Ch. 6) linear classifiers to fine-tune the category vectors. Wang et al. (2003) used Medical Subject Headings (MeSH) (MeSH, 2003) to replace the bag of words with canonical medical terms; Bloehdorn and Hotho (2004) used a similar approach to augment Reuters-21578 documents with WordNet synsets and OHSUMED medical documents with MeSH terms.

It should be noted, however, that WordNet was not originally designed to be a powerful knowledge base, but rather a lexical database more suitable for peculiar lexicographers' needs. Specifically, WordNet has the following drawbacks when used as a knowledge base for text categorization:

- WordNet has a fairly small coverage—for the test collections we used in this paper, up to 50% of their unique words are missing from WordNet. In particular, many proper names, slang and domain-specific technical terms are not included in WordNet, which was designed as a general-purpose dictionary.

- Additional information about synsets (beyond their identity) is very limited. This is because WordNet implements a *differential* rather than *constructive* lexical semantics theory, so that glosses that accompany the synsets are mainly designed to distinguish the synsets rather than provide a definition of the sense or concept. Usage examples that occasionally constitute part of the gloss serve the same purpose. Without such auxiliary information, reliable word sense disambiguation is almost impossible.

- WordNet was designed by professional linguists who are trained to recognize minute differences in word senses. As a result, common words have far too many distinct senses to be useful in information retrieval (Mihalcea, 2003); for example, the word "make" has as many as 48 senses as a verb alone. Such fine-grained distinctions between synsets present an additional difficulty for word sense disambiguation.

We illustrate these drawbacks on two specific examples in Appendix B, where we juxtapose WordNet-based and ODP-based feature generation.

The methodology we propose in this paper does not suffer from the above shortcomings. Crawling all the Web sites cataloged in the Open Directory results in exceptionally wide word coverage. Furthermore, the crawled texts provide a plethora of information about each ODP concept.

To the best of our knowledge, with the exception of the above WordNet studies, there have been no attempts to date to automatically use large-scale repositories of structured background knowledge for text categorization. An interesting approach to using non-structured background knowledge was proposed by Zelikovitz and Hirsh (2000). This work

[13] A *synset* is WordNet notion for a sense shared by a group of synonymous words.

uses a collection of unlabeled examples as intermediaries in comparing testing examples with the training ones. Specifically, when an unknown test instance does not appear to resemble any labeled training instances, unlabeled examples that are similar to both may be used as "bridges." Using this approach, it is possible to handle the situation where the training and the test document have few or no words in common. The unlabeled documents are utilized to define a cosine similarity metric, which is then used by the KNN algorithm for actual text categorization. This approach, however, suffers from efficiency problems, as looking for intermediaries to compare every two documents makes it necessary to explore a combinatorial search space. In a subsequent paper, Zelikovitz and Hirsh (2001) proposed an alternative way to use unlabeled documents as background knowledge. In this work, unlabeled texts are pooled together with the training documents to compute a Latent Semantic Analysis (LSA) (Deerwester et al., 1990) model. The resulting LSA metric then facilitates comparison of test documents to training documents. The addition of unlabeled documents significantly increases the amount of data on which word cooccurrence statistics is estimated, thus providing a solution to text categorization problems where training data is particularly scarce.

The methodology described in this paper uses external knowledge explicitly cataloged by humans to enhance machine learning algorithms. There have also been other studies (notably, using semi-supervised learning methodology) that augmented the bag of words approach to text categorization with external knowledge distilled from unlabelled data (Goldberg and Zhu, 2006; Ando and Zhang, 2005a,b; Blei et al., 2003; Nigam et al., 2000; Joachims, 1999b). Consequently, it would be very interesting to compare the performance of these two approaches empirically. Intuitively, some inferences, such as those described in Section 4, would be hard to make by using solely unstructured data. On the other hand, unstructured data is more readily available, so it is possible that semi-supervised methods can compensate for the lack of structure by increasing the volume of the data. There are, however, non-trivial research questions regarding an appropriate setup for such a comparison. For example, assuming our methodology is based on the ODP as described in this paper, what corpus should be used by the semi-supervised learner? And if one of the methods shows better performance, should it be attributed to the method or to the particular knowledge source being used? We plan to investigate these and other related questions in our future work. In any case, it is most likely that each of the methods has its own strengths, and finding a way to combine them can be a very interesting research direction.

While our approach relies on existing repositories of classified knowledge, there is a large body of research on extracting facts through Web mining (Cafarella et al., 2005; Etzioni et al., 2004), so it would be interesting to consider using such extracted facts to drastically increase the amount of available knowledge, especially when measures are taken to ascertain correctness of the extracted information (Downey et al., 2005).

In a recent study (Gabrilovich and Markovitch, 2007), we applied our methodology to the problem of computing semantic relatedness of words and texts, for which previous state of the art results have been based on LSA. In that work we proposed Explicit Semantic Analysis (ESA), which represents fragments of text in the space of knowledge concepts defined in the Open Directory or in Wikipedia. ESA uses the same basic feature generation methodology that we presented herein, but represents texts in the space of all available concepts (discarding the bag of words altogether), rather then augmenting the bag of words with a few top scoring concepts. Compared with the existing state of the art, using ESA results in substantial improvements in correlation of computed relatedness scores with human judgments: from $r = 0.56$ to 0.75 for individual words and from $r = 0.60$ to 0.72 for longer texts. These findings prove that the benefits of using distilled human knowledge are much greater than merely using cooccurrence statistics gathered from a collection of auxiliary unlabeled texts.

Our use of local contexts to facilitate fine-grained feature generation is reminiscent of the intra-document dynamics analysis proposed by Gabrilovich et al. (2004) for characterization of news article types. The latter work manipulated sliding contextual windows of the same size to make their scores directly comparable. As we showed in Section 5.5, the multi-resolution approach, which operates at several levels of linguistic abstraction, is superior to fixed-size windows for the case of text categorization. Incidentally, the term "Local Context Analysis" is also used in an entirely different branch of information retrieval. Xu and Croft (2000) used this term to refer to a particular kind of query expansion, where a query is expanded *in the context* of top-ranked retrieved documents.

In our methodology, we first learn a text classifier that maps local document contexts onto ODP concepts, and then use this classifier for feature generation in other learning tasks. This framework is clearly related to the area of *transfer learning*, where knowledge learned in one domain is transferred to another domain. Some works in this area assume a set of related classification tasks, and learn shared parameters. For example, Caruana (1997) trained one neural network for several related classification tasks, such that nodes in the hidden level were useful across the tasks. Jebara (2004) presented a method for feature and kernel selection across related tasks. Do and Ng (2005) described a general way of using softmax regression for learning a parameter function from a set of classification problems, so that the learned parameter function can then be used for future learning tasks. Bennett et al. (2005) introduced a method for learning a meta-classifier over several domains. The meta-classifier combines reliability indicators (Toyama and Horvitz, 2000) with the base classifiers to improve their performance. Several studies in NLP Sutton and McCallum (1998); Chang et al. (2006); Raina et al. (2007); Ando and Zhang (2005a) and image classification Wu and Dietterich (2004); Raina et al. (2007); Ando and Zhang (2005a) used a cascade approach, where a classifier trained on one task is used as a feature for another task. This type of transfer is the most similar to ours, as we also use a very large set of such classifiers trained on the ODP as features in other learning tasks.

7. Conclusions and Future Work

In this paper we proposed a feature generation methodology for text categorization. In order to render machine learning algorithms with the common-sense and domain-specific knowledge of humans, we use large hierarchical knowledge repositories to build a *feature generator*. These knowledge repositories, which have been manually crafted by human editors, provide a fully automatic way to tap into the collective knowledge of tens of thousands of people. The feature generator analyzes documents prior to text categorization and augments the conventional bag of words representation with relevant concepts from the knowledge repository. The enriched representation contains information that cannot be deduced from the document text alone.

In Section 2 we listed several limitations of the BOW approach, and in the subsequent sections we showed how they are resolved by our methodology. In particular, external knowledge allows us to reason about words that appear in the testing set but not in the training set. We can also use hierarchically organized knowledge to make powerful generalizations, making it possible to know that certain infrequent words belong to more general classes of words. Externally supplied knowledge can also help in those cases when some information vital for classification is omitted from training texts because it is assumed to be shared by the target readership.

We also described multi-resolution analysis, which examines the document text at several levels of linguistic abstraction and performs feature generation at each level. When polysemous words are considered in their native context, word sense disambiguation is implicit. Implicit disambiguation allows the feature generator to cope with word synonymy and polysemy. Furthermore, when the document text is processed at several levels of granularity, even briefly mentioned aspects can be identified and used. These might easily have been overlooked if the document were processed as one large chunk of text.

Empirical evaluation definitively confirmed that knowledge-based feature generation brings text categorization to a new level of performance. Interestingly, the sheer breadth and depth of the ODP, further boosted by crawling the URLs cataloged in the directory, brought about improvements both in regular text categorization as well as in the (non-topical) sentiment classification task.

Given the domain-specific nature of some test collections, we also compared the utility of narrow domain-specific knowledge with that of larger amounts of information covering all branches of knowledge. Perhaps surprisingly, we found that even for narrow-scope test collections, a wide coverage knowledge base yielded substantially greater improvements than its domain-specific subsets. This observation reinforces the *breadth hypothesis*, formulated by Lenat and Feigenbaum (1990), that “to behave intelligently in unexpected situations, an agent must be capable of falling back on increasingly general knowledge.”

We believe that this research only scratches the surface of what can be achieved with knowledge-rich features. In our future work, we plan to investigate new algorithms for mapping document contexts onto hierarchy concepts, as well as new techniques for selecting attributes that are most characteristic of every concept. We intend to apply focused crawling to collect only relevant Web pages when cataloged URLs are crawled; we also plan to apply page segmentation techniques to eliminate noise from crawled pages Yu et al. (2003). In addition to the ODP, we also plan to make use of domain-specific hierarchical knowledge bases, such as the Medical Subject Headings (MeSH).

In its present form, our method can inherently be applied only for improving representation of textual documents. Indeed, to date we applied our feature generation methodology for improving the performance of text categorization. However, we believe our approach can also be applied beyond mere text, as long as the objects to be manipulated are accompanied with some textual description. As an example, consider a collection of medical records containing test results paired with narrative reports. Performing feature generation from narrative reports is likely to produce pertinent concepts that can be used for augmenting the original record. Indeed, prior studies (Hripcsak et al., 1995) showed that natural language processing techniques can be used to extract vital information from narrative reports in automated decision-support systems.

Finally, we conjecture that knowledge-based feature generation will also be useful for other information retrieval tasks beyond text categorization, and we intend to investigate this in our future work. Specifically, we intend to apply feature generation to information search and word sense disambiguation. In the search scenario, we are studying ways to augment both the query and documents in the collection with generated features. This way, documents will be indexed in the augmented space of words plus concepts. In this respect, we are exploring possible use of relevance feedback techniques (Ruthven and Lalmas, 2003) in order to augment the query with most useful generated features. Current approaches to word sense disambiguation represent contexts that contain ambiguous words using the bag of words augmented with part-of-speech information. To this end, we believe representation of such contexts can be greatly improved if we use feature generation to map such contexts into relevant knowledge concepts. Anecdotal evidence (such as the last example in Section 5.3.1) implies our method has much promise for improving the state of the art in word sense disambiguation.

Acknowledgments

We thank Lev Finkelstein and Alex Gontmakher for many helpful discussions. This research was partially supported by the Technion's Counter-Terrorism Competition and by the MUSCLE Network of Excellence.

References

Rie Kubota Ando and Tong Zhang. Framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, pages 1817–1853, 2005a.

Rie Kubota Ando and Tong Zhang. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 1–9, Ann Arbor, MI, June 2005b.

Douglas Baker and Andrew K. McCallum. Distributional clustering of words for text classification. In Bruce Croft, Alistair Moffat, Cornelis J. Van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US. URL `http://www.cs.cmu.edu/ mccallum/papers/clustering-sigir98.ps.gz`.

Lisa Ballesteros and Bruce Croft. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th ACM International Conference on Research and Development in Information Retrieval*, pages 84–91, 1997.

Roberto Basili, Alessandro Moschitti, and Maria T. Pazienza. Language-sensitive text classification. In *Proceedings of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur"*, pages 331–343, Paris, France, 2000.

Ron Bekkerman. Distributional clustering of words for text categorization. Master's thesis, Technion, 2003.

Paul N. Bennett, Susan T. Dumais, and Eric Horvitz. Inductive transfer for text classification using generalized reliability indicators. In *Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, 2003.

Paul N. Bennett, Susan T. Dumais, and Eric Horvitz. The combination of text classifiers using reliability indicators. *Information Retrieval*, 8(1):67–100, 2005.

Cathy Blake and Christopher Merz. UCI Repository of machine learning databases, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent direchlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Stephan Bloehdorn and Andreas 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*, pages 70–87, 2004.

Janez Brank, Marko Grobelnik, Natasa Milic-Frayling, and Dunia Mladenic. Interaction of feature selection methods and linear classification models. In *Workshop on Text Learning held at ICML-2002*, 2002.

Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

Michael Cafarella, Doug Downey, Stephen Soderland, and Oren Etzioni. Knowitnow: Fast, scalable information extraction from the web. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Vancouver, Canada, October 2005.

Lijuan Cai and Thomas Hofmann. Text categorization by boosting automatically extracted concepts. In *Proceedings of the 26th International Conference on Research and Development in Information Retrieval*, pages 182–189, 2003.

Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Amita G. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, US, 2001. URL `http://faure.iei.pi.cnr.it/ fabrizio/Publications/TD01a/TD01a.pdf`.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. Using taxonomy, discriminants, and signatures for navigating in text databases. In *Proceedings of the 23rd VLDB Conference*, pages 446–455, 1997.

Lois Mai Chan. *A Guide to the Library of Congress Classification.* Libraries Unlimited, 5th edition, 1999.

Ming-Wei Chang, Quang Do, and Dan Roth. Multilingual dependency parsing: A pipeline approach. In *Recent Advances in Natural Language Processing*, pages 195–204, 2006.

Stanley Chen and Joshua Goodman. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 34th Annual Meeting of the ACL*, 1996.

William W. Cohen. Automatically extracting features for concept learning from the web. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.

Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2/3):127–152, 2002.

Dmitry Davidov, Evgeniy Gabrilovich, and Shaul Markovitch. Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In *Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval*, pages 250–257, 2004.

Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing*, pages 784–788, 2003.

Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Gerald Dejong and Raymond Mooney. Explanation-based learning: An alternative view. *Machine Learning*, 1(2):145–176, 1986.

Melvil Dewey, Joan S. Mitchell, Julianne Beall, Giles Martin, Winton E. Matthews, and Gregory R. New, editors. *Dewey Decimal Classification and Relative Index.* Online Computer Library Center (OCLC), 22nd edition, 2003.

Inderjit Dhillon, Subramanyam Mallela, and Rahul Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, March 2003. URL `http://www.jmlr.org/papers/volume3/dhillon03a/dhillon03a.pdf`.

Chuong Do and Andrew Ng. Transfer learning for text classification. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2005.

Doug Downey, Oren Etzioni, and Stephen Soderland. A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotand, August 2005.

Richard Duda and Peter Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

Susan Dumais and Hao Chen. Hierarchical classification of web content. In *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, 2000.

Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, 1998.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel Weld, and Alexander Yates. Webscale information extraction in knowitall (preliminary results). In *Proceedings of the 13th International World Wide Web Conference (WWW'04)*, New York, USA, May 2004. ACM Press.

Tom Fawcett. *Feature Discovery for Problem Solving Systems*. PhD thesis, UMass, May 1993.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

Johannes Fuernkranz, Tom Mitchell, and Ellen Riloff. A case study in using linguistic phrases for text categorization on the WWW. In Mehran Sahami, editor, *Learning for Text Categorization: Proceedings of the 1998 AAAI/ICML Workshop*, pages 5–12. AAAI Press, Madison, Wisconsin, 1998.

Evgeniy Gabrilovich and Shaul Markovitch. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In *Proceedings of the 21st International Conference on Machine Learning*, pages 321–328, 2004.

Evgeniy Gabrilovich and Shaul Markovitch. Feature generation for text categorization using world knowledge. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1048–1053, Edinburgh, Scotand, August 2005.

Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1301–1306, July 2006.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, January 2007.

Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW2004)*, pages 482–490, New York, NY, May 2004. ACM Press.

Andrew Goldberg and Xiaojin Zhu. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing, HLT-NAACL 2006*, 2006.

Eui-Hong (Sam) Han and George Karypis. Centroid-based document classification: Analysis and experimental results. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, September 2000.

William Hersh, Chris Buckley, T.J. Leone, and David Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, pages 192–201, 1994.

George Hripcsak, Carol Friedman, Philip O. Alderson, William DuMouchel, Stephen B. Johnson, and Paul D. Clayton. Unlocking clinical data from narrative reports: a study of natural language processing. *Annals of Internal Medicine*, 122(9):681–688, 1995.

Yuh-Jyh Hu and Dennis Kibler. A wrapper approach for constructive induction. In *The Thirteenth National Conference on Artificial Intelligence*, pages 47–52, 1996.

David A. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In W. Bruce Croft and Cornelis J. Van Rijsbergen, editors, *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, pages 282–289, Dublin, Ireland, 1994. Springer Verlag, Heidelberg, Germany. URL `http://www.acm.org/pubs/articles/proceedings/ir/188490/p282-hull/p282-hull.pdf`.

Tony Jebara. Multi-task feature and kernel selection for svms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 55–63, 2004.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pages 137–142, 1998.

Thorsten Joachims. Making large-scale SVM learning practical. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 169–184. The MIT Press, 1999a.

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 13th International Conference on Machine Learning*, 1999b.

Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning*, pages 170–178, 1997.

Daniel Kudenko and Haym Hirsh. Feature generation for sequence categorization. In *Proceedings of the 15th Conference of the American Association for Artificial Intelligence*, pages 733–738, 1998.

Giridhar Kumaran and James Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval*, pages 297–304, 2004.

Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339, 1995.

Douglas Lenat and Edward Feigenbaum. On the thresholds of knowledge. *Artificial Intelligence*, 47:185–250, 1990.

Edda Leopold and Joerg Kindermann. Text categorization with support vector machines: How to represent texts in input space. *Machine Learning*, 46:423–444, 2002.

David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, 1992.

David D. Lewis and W. Bruce Croft. Term clustering of syntactic phrases. In *Proceedings of the 13th ACM International Conference on Research and Development in Information Retrieval*, pages 385–404, 1990.

David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In *Proceedings of the 19th ACM International Conference on Research and Development in Information Retrieval*, pages 298–306, 1996.

David D. Lewis, Yiming Yang, Tony Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Shaul Markovitch and Danny Rosenstein. Feature generation using general constructor functions. *Machine Learning*, 49(1):59–98, 2002.

Christopher J. Matheus. The need for constructive induction. In L.A. Birnbaum and G.C. Collins, editors, *Proceedings of the Eighth International Workshop on Machine Learning*, pages 173–177, 1991.

Christopher J. Matheus and Larry A. Rendell. Constructive induction on decision trees. In *Proceedings of the 11th International Conference on Artificial Intelligence*, pages 645–650, 1989.

Ia Mcilwaine. *The Universal Decimal Classification: Guide to its Use.* UDC Consortium, 2000.

MeSH. Medical subject headings (MeSH). National Library of Medicine, 2003. `http://www.nlm.nih.gov/mesh`.

Rada Mihalcea. Turning wordnet into an information retrieval resource: Systematic polysemy and conversion to hierarchical codes. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 17(1):689–704, 2003.

Andrei Mikheev. Feature lattices and maximum entropy models. *Information Retrieval*, 1999.

Tom Mitchell, Richard Keller, and Smadar Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47–80, 1986.

Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval*, pages 206–214, 1998.

Dunja Mladenic. Feature subset selection in text learning. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 95–100, 1998a.

Dunja Mladenic. Turning Yahoo into an automatic web-page classifier. In *Proceedings of 13th European Conference on Artificial Intelligence*, pages 473–474, 1998b.

Patrick M. Murphy and Michael J. Pazzani. ID2-of-3: Constructive induction of M-of-N concepts for discriminators in decision trees. In *Proceedings of the 8th International Conference on Machine Learning*, pages 183–188. Morgan Kaufmann, 1991.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000.

Kamal Nigam, Andrew McCallum, and Tom Mitchell. Semi-supervised text classification using EM. In Olivier Chapelle, Bernhard Schoelkopf, and Alexander Zien, editors, *Semi-Supervised Learning*. MIT Press, Boston, MA, 2006.

Giulia Pagallo and David Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 5(1):71–99, 1990. ISSN 0885-6125.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–86, 2002.

Fuchun Peng and Dale Shuurmans. Combining naive Bayes and n-gram language models for text classification. In *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR-03)*, pages 335–350, 2003.

Fuchun Peng, Dale Schuurmans, and Shaojun Wang. Augmenting naive Bayes classifiers with statistical language models. *Information Retrieval*, 7(3-4):317–345, 2004.

Martin Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

Rajat Raina, Andrew Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, Pittsburgh, PA, 2006.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML '07: Proceedings of the 24th International Conference on Machine learning*, 2007.

Bhavani Raskutti, Herman Ferra, and Adam Kowalczyk. Second order features for maximizing text classification performance. In L. De Raedt and P. Flach, editors, *Proceedings of the European Conference on Machine Learning (ECML)*, Lecture notes in Artificial Intelligence (LNAI) 2167, pages 419–430. Springer-Verlag, 2001.

Reuters. *Reuters-21578 text categorization test collection, Distribution 1.0.* Reuters, 1997. daviddlewis.com/resources/testcollections/reuters21578.

Joseph John Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

Monica Rogati and Yiming Yang. High-performing feature selection for text classification. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'02)*, pages 659–661, 2002.

J.K. Rowling. *Harry Potter and the Philosopher's Stone*. Bloomsbury, 1997.

Miguel E. Ruiz and Padmini Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5:87–118, 2002.

Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145, 2003.

Carl Sable, Kathleen McKeown, and Kenneth W. Church. NLP found helpful (at least for one text categorization task). In *Conference on Empirical Methods in Natural Language Processing*, pages 172–179, 2002.

Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

Gerard Salton and Michael McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

Sam Scott. Feature engineering for a symbolic approach to text classification. Master's thesis, U. Ottawa, 1998.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. URL `http://faure.iei.pi.cnr.it/ fabrizio/Publications/ACMCS02.pdf`.

Charles Sutton and Andrew McCallum. Composition of conditional random fields for transfer learning. In *Emprical Methods in Natural Language Processing (HLT/EMNLP)*, 1998.

Kentaro Toyama and Eric Horvitz. Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. In *Proceedings of the 4th Asian Conference on Computer Vision*, 2000.

Alfonso Urena-Lopez, Manuel Buenaga, and Jose M. Gomez. Integrating linguistic resources in TC through WSD. *Computers and the Humanities*, 35:215–230, 2001.

Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, pages 61–69, 1994.

Ellen M. Voorhees. Using wordnet for text retrieval. In Christiane Fellbaum, editor, *WordNet, an Electronic Lexical Database*. The MIT Press, 1998.

Bill B. Wang, R.I. McKay, Hussein A. Abbass, and Michael Barlow. A comparative study for domain ontology guided feature extraction. In *Proceedings of the 26th Australian Computer Science Conference (ASCS-2003)*, pages 69–78, 2003.

Bernard Widrow and Samuel Stearns. *Adaptive Signal Processing*. Prentice Hall, 1985.

Michael Wong, Wojciech Ziarko, and Patrick C.N. Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th ACM International Conference on Research and Development in Information Retrieval*, pages 18–25, 1985.

Pengcheng Wu and Thomas G. Dietterich. Improving svm accuracy by training on auxiliary data sources. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 871–878, New York, NY, USA, 2004. ACM Press.

Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.

Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, 2000.

Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.

Yiming Yang and Jan Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, 1997.

Yiming Yang, Sean Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2/3):219–241, 2002.

Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th International World Wide Web Conference (WWW'03)*, Budapest, Hungary, May 2003. ACM Press.

Sarah Zelikovitz and Haym Hirsh. Improving short-text classification using unlabeled background knowledge to assess document similarity. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1183–1190, 2000.

Sarah Zelikovitz and Haym Hirsh. Using LSI for text classification in the presence of background text. In *Proceedings of the Conference on Information and Knowledge Management*, pages 113–118, 2001.

Justin Zobel and Alistair Moffat. Exploring the similarity space. *ACM SIGIR Forum*, 32 (1):18–34, 1998.

Appendix A. Definitions of category sets for RCV1 and OHSUMED

This Appendix gives the full definition of the category sets we used for RCV1 (Table 8) and OHSUMED (Table 9).

Set name	Categories comprising the set
Topic-16	e142, gobit, e132, c313, e121, godd, ghea, e13, c183, m143, gspo, c13, e21, gpol, m14, c15
Topic-10A	e31, c41, c151, c313, c31, m13, ecat, c14, c331, c33
Topic-10B	m132, c173, g157, gwea, grel, c152, e311, c21, e211, c16
Topic-10C	c34, c13, gtour, c311, g155, gdef, e21, genv, e131, c17
Topic-10D	c23, c411, e13, gdis, c12, c181, gpro, c15, g15, c22
Topic-10E	c172, e513, e12, ghea, c183, gdip, m143, gcrim, e11, gvio
Industry-16	i81402, i79020, i75000, i25700, i83100, i16100, i1300003, i14000, i3302021, i8150206, i0100132, i65600, i3302003, i8150103, i3640010, i9741102
Industry-10A	i47500, i5010022, i3302021, i46000, i42400, i45100, i32000, i81401, i24200, i77002
Industry-10B	i25670, i61000, i81403, i34350, i1610109, i65600, i3302020, i25700, i47510, i9741110
Industry-10C	i25800, i41100, i42800, i16000, i24800, i02000, i34430, i36101, i24300, i83100
Industry-10D	i1610107, i97400, i64800, i0100223, i48300, i81502, i34400, i82000, i42700, i81402
Industry-10E	i33020, i82003, i34100, i66500, i1300014, i34531, i16100, i22450, i22100, i42900

Table 8: Definition of RCV1 category sets used in the experiments

Appendix B. Comparing knowledge sources for feature generation: ODP versus WordNet

In Section 6 we surveyed the shortcomings of WordNet as a possible source for knowledge-based feature generation. To demonstrate these shortcomings, we juxtapose WordNet-based and ODP-based feature generation for two of sample sentences we examined in Section 5.3.1 (we repeat the ODP context classifications for readers' convenience). We used WordNet version 1.6 to look up the words. In what follows, synsets are denoted with curly braces, and noun and verb synsets are followed by their immediate hypernym (more general synset), if applicable.

- **Text:** *"Rumsfeld appeared with Gen. Richard Myers, chairman of the Joint Chiefs of Staff."*

  ODP classifications:

  - Society/Issues/Government_Operations
  - Society/Politics
  - Society/Issues/Warfare_and_Conflict/Specific_Conflicts/Iraq
  - Science/Technology/ Military_Science
  - Society/Issues/Warfare_and_Conflict/Weapons

Set name	**Categories comprising the set** (parentheses contain MeSH identifiers)
OHSUMED-10A	B-Lymphocytes (D001402); Metabolism, Inborn Errors (D008661); Creatinine (D003404); Hypersensitivity (D006967); Bone Diseases, Metabolic (D001851); Fungi (D005658); New England (D009511); Biliary Tract (D001659); Forecasting (D005544); Radiation (D011827)
OHSUMED-10B	Thymus Gland (D013950); Insurance (D007341); Historical Geographic Locations (D017516); Leukocytes (D007962); Hemodynamics (D006439); Depression (D003863); Clinical Competence (D002983); Anti-Inflammatory Agents, Non-Steroidal (D000894); Cytophotometry (D003592); Hydroxy Acids (D006880)
OHSUMED-10C	Endothelium, Vascular (D004730); Contraceptives, Oral, Hormonal (D003278); Acquired Immunodeficiency Syndrome (D000163); Gram-Positive Bacteria (D006094); Diarrhea (D003967); Embolism and Thrombosis (D016769); Health Behavior (D015438); Molecular Probes (D015335); Bone Diseases, Developmental (D001848); Referral and Consultation (D012017)
OHSUMED-10D	Antineoplastic and Immunosuppressive Agents (D000973); Receptors, Antigen, T-Cell (D011948); Government (D006076); Arthritis, Rheumatoid (D001172); Animal Structures (D000825); Bandages (D001458); Italy (D007558); Investigative Techniques (D008919); Physical Sciences (D010811); Anthropology (D000883)
OHSUMED-10E	HTLV-BLV Infections (D006800); Hemoglobinopathies (D006453); Vulvar Diseases (D014845); Polycyclic Hydrocarbons, Aromatic (D011084); Age Factors (D000367); Philosophy, Medical (D010686); Antigens, CD4 (D015704); Computing Methodologies (D003205); Islets of Langerhans (D007515); Regeneration (D012038)

Table 9: Definition of OHSUMED category sets used in the experiments

- Society/History/By_Region/North_America/United_States/Presidents/Bush,_George_Walker
- Society/Politics/Conservatism

WordNet :
{**Rumsfeld**} → { }; *(word not present in WordNet)*

{look, **appear**, seem} → {be}; {**appear**}; {**appear**, come out} → {happen, materialize}; {**appear**, seem} → {be}; {**appear**, come along}; {**appear**} → {perform, execute, do}

{**Gen**} → {information, info}

{**Richard**} → { }; *(word not present in WordNet)*

{**Myers**} → { }; *(word not present in WordNet)*

{president, **chairman**, chairwoman, chair, chairperson} → {presiding officer}; {chair, **chairman**} → {head, lead}

{**joint**, articulation, articulatio} → {body part}; {**joint**} → {spot}; {articulation, join, **joint**, juncture, junction} → {connection, connexion, link}; {roast, **joint**} → {cut, cut of meat}; {**joint**} → {junction, conjunction}; {**joint**, marijuana cigarette, reefer, stick} → {cigarette, cigaret, coffin nail, butt, fag}

{**joint**} → {fit, go}; {**joint**, articulate} → {supply, provide, render, furnish}; {**joint**} → {fasten, fix, secure}

{**joint** (vs. separate)}; {joint}

{head, **chief**, top dog} → {leader}; {foreman, **chief**, gaffer, honcho, boss} → {supervisor}

{**staff**} → {force, personnel}; {**staff**} → {stick}; {**staff**, faculty} → {body}; {**staff**} → {symbol}; {**staff**, stave} → {musical notation}

{**staff**} → {provide, supply, ply, cater}

- **Text:** *"Herceptin is a so-called targeted therapy because of its ability to attack diseased cells and leave healthy ones alone."*

  ODP classifications:

  - Health/Conditions_and_Diseases/Cancer/Breast
  - Society/Issues/Health/Conditions_and_Diseases/Cancer/Alternative_Treatments
  - Health/Support_Groups/Conditions_and_Diseases/Cancer

  WordNet:

  {**Herceptin**} → { }; *(word not present in WordNet)*

  {alleged (prenominal), **so-called**, supposed} → {questionable (vs. unquestionable)}

  {**target**, aim, place, direct, point} → {aim, take, train, take aim, direct} {**therapy**} → {medical care, medical aid}

{**ability**} → {quality}

{**ability**, power} → {cognition, knowledge}

{**attack**, onslaught, onset, onrush} → {operation}; {**attack**} → {turn, play}; {fire, **attack**, flak, blast} → {criticism, unfavorable judgment}; {approach, **attack**, plan of attack} → { conceptualization, conceptualisation, formulation, formularizing, formularising}; {**attack**, attempt} → {battery, assault, assault and battery}; {**attack**, tone-beginning} → {beginning, start, commencement}; {**attack**} → {affliction}; {**attack**, assault} → {attention, attending};

{**attack**, assail} → {fight, struggle}; {**attack**, round, assail, lash out, snipe, assault} → {criticize, criticise, pick apart}; {**attack**, aggress} → {act, move}; {assail, assault, set on, **attack**}; {**attack**} → {begin, get, start out, start, set about, set out, commence}; {**attack**} → {affect}

{assault (prenominal), **attack** (prenominal)} → {offensive (vs. defensive)};

{**diseased**, morbid, pathologic, pathological} → {unhealthy (vs. healthy)};

{**cell**} → {compartment}; {**cell**} → {entity, something}; {**cell**, electric cell} → {electrical device}; {**cell**, cadre} → {political unit}; {**cell**, cubicle} → {room}; {**cell**, jail cell, prison cell} → {room}

{**leave**, leave of absence} → {time off}; {**leave**} → {permission}; {farewell, **leave**, leave-taking, parting} → {departure, going, going away, leaving};

{**leave**, go forth, go away}; *(16 more verb senses omitted for brevity)*

{**healthy** (vs. unhealthy)}; {**healthy**} → {sound (vs. unsound)}; {**healthy**, salubrious, good for you (predicate)} → {wholesome (vs. unwholesome)}; {fit (vs. unfit), **healthy**} → {able, able-bodied}; {**healthy**, intelligent, levelheaded, sound} → {reasonable (vs. unreasonable), sensible};

{**one**, 1, I, ace, single, unity} → {digit}; {**one**} → {unit}

{**alone** (predicate)} → {unsocial (vs. social)}; {**alone** (predicate), lone (prenominal), lonely (prenominal), solitary} → {unaccompanied (vs. accompanied)}; {**alone** (predicate), only} → {exclusive (vs. inclusive)}; {**alone** (predicate), unique, unequaled, unequalled, unparalleled} → {incomparable (vs. comparable), uncomparable}

{entirely, exclusively, solely, **alone**, only}; {**alone**, unaccompanied}

Evidently, WordNet classifications are overly general and diverse because context words cannot be properly disambiguated. Furthermore, owing to lack of proper names, WordNet cannot possibly provide the wealth of information encoded in the Open Directory, which easily overcomes the drawbacks of WordNet.