Automatic Expansion of Domain-Specific Lexicons by Term Categorization

HENRI AVANCINI
Consiglio Nazionale delle Ricerche, Italy
ALBERTO LAVELLI
ITC-irst, Italy
FABRIZIO SEBASTIANI
Consiglio Nazionale delle Ricerche, Italy
ROBERTO ZANOLI
ITC-irst, Italy

We discuss an approach to the automatic expansion of *domain-specific lexicons*, that is, to the problem of extending, for each c_i in a predefined set $C = \{c_1, \ldots, c_m\}$ of semantic *domains*, an initial lexicon L_0^i into a larger lexicon L_1^i. Our approach relies on *term categorization*, defined as the task of labeling previously unlabeled terms according to a predefined set of domains. We approach this as a supervised learning problem in which term classifiers are built using the initial lexicons as training data. Dually to classic text categorization tasks in which documents are represented as vectors in a space of terms, we represent terms as vectors in a space of documents. We present the results of a number of experiments in which we use a boosting-based learning device for training our term classifiers. We test the effectiveness of our method by using WordNetDomains, a well-known large set of domain-specific lexicons, as a benchmark. Our experiments are performed using the documents in the Reuters Corpus Volume 1 as implicit representations for our terms.

Categories and Subject Descriptors: I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms: Algorithms, Experimentation, Theory

Additional Key Words and Phrases: Lexicons, text classification, machine learning

This work has been carried out in the framework of the WebFAQ project, funded by the Provincia Autonoma di Trento.

Authors' addresses: H. Avancini, F. Sebastiani, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via Giuseppe Moruzzi 1, 56124 Pisa, Italy; e-mail: {henri.avancini,fabrizio.sebastiani}@isti.cnr.it. A. Lavelli, R. Zanoli, ITC-irst, Via Sommarive 18, 38050 Povo (TN), Italy; e-mail: {lavelli,zanoli}@itc.it. Address all correspondence to F. Sebastiani.

1. INTRODUCTION

The generation of *domain-specific lexicons* (i.e., lexicons consisting of terms pertaining to a given domain or discipline) is a task of increased applicative interest since such lexicons are of the utmost importance in a variety of tasks pertaining to natural language processing and information access. One of these tasks is, for instance, query expansion for information retrieval (IR) systems addressing specialized document collections (as in, e.g., thematic, "vertical" portals) in which terms synonymous or quasisynonymous to the query terms are added to the query in order to retrieve more relevant documents. Domain-specific lexicons are also useful for *word-sense disambiguation* (WSD), the task of determining, given the occurrence o_w of a polysemous word w, the sense of o_w. Results from Magnini et al. [2002] indicate that, given a word occurrence o_w whose possible senses $w_1, \ldots, w_s$ pertain each to a different domain $d_1, \ldots, d_s$, the domain d_i to which most of the terms occurring in the context of o_w pertain has a high probability of indicating the correct sense; that is, there is a high probability that the right sense of o_w is w_i (see Figure 1). Domain-specific lexicons are then of fundamental importance for WSD since it is important to know to which domains the terms occurring in the context of o_w pertain. Thelen and Riloff [2002] quote several other works in which domain-specific lexicons (which they call *semantic lexicons*) have proven useful for performing natural language processing tasks such as information extraction [Riloff and Shepherd 1999; Soderland et al. 1995], anaphora resolution [Aone and Bennett 1996], question answering [Moldovan et al. 1999; Hirschman et al. 1999], and prepositional phrase attachment [Brill and Resnik 1994].

Unfortunately, the manual generation of domain-specific lexicons is expensive, since it requires the intervention of human experts, that is, lexicographers and domain experts working together. Besides being expensive, such a manual approach does not allow for fast response to rapidly emerging needs; in an era of frantic technical progress, new disciplines emerge quickly, while others disappear as quickly, and in an era of evolving consumer needs, the same goes for new market niches. There is thus a need for cheaper and faster methods for answering application needs than manual lexicon generation. The manual approach is also prone to errors of omission in that a lexicographer may easily overlook infrequent, nonobvious terms that are nonetheless important for many tasks.

Many applications also require that the lexicons be not only domain-specific, but also tailored to the specific data tackled in the application. For instance, in query expansion for IR systems addressing specialized document collections, the synonymous or quasisynonymous terms to be added to the query terms should be chosen among the ones that occur in the document collection, otherwise they would be useless; conversely, the relevant terms which occur in the document collection should potentially be added. Therefore, for this application the ideal domain-specific lexicon should contain all and only the technical terms that occur in the document collection under consideration and should thus be generated directly from this collection.

From the plush Connolly hide leather [sofa]$_F$ and [chairs]$_F$ in the [living room]$_F$ to the [Bang and Olufsen stereo]$_F$, and [remote control television]$_F$ complete with video, you're surrounded by the HIGHEST QUALITY. The [inlaid]$_F$ chequerboard top of the [coffee table]$_F$ houses all kind of [games]$_P$, including [backgammon]$_P$, [chess]$_P$ and [Scrabble]$_P$. You'll also find a selection of books, from Queen Victoria's Highland journals, to the very latest bestselling [thriller]$_P$. The [dinner table]$_F$ and [**chairs**]$_{??}$ are elegant yet comfortable, and you can be assured of the finest [tableware]$_F$ and crystal for meals at home.

Fig. 1. Word sense disambiguation by using domain information (example taken from Senseval-2, the international campaign for the experimental evaluation of WSD systems). Subscripts appended to terms indicate the domain to which the terms are known to belong (F=FURNITURE, P=PLAY, L=LITERATURE). The word occurrence to be disambiguated is the second occurrence of the word "chairs" and its possible senses pertain to the domains LITERATURE, FURNITURE, and PLAY. Since most of the terms occurring in the context of this word occurrence belong to the FURNITURE domain, it seems reasonable to conclude that also this occurrence of "chairs" should belong to FURNITURE.

1.1 Our Proposal

In this article, we propose a methodology for the automatic expansion of domain-specific lexicons. This methodology relies on *term categorization*, a novel task that employs a combination of techniques from IR and machine learning (ML). Specifically, we view the expansion of such lexicons as a process of learning, from a corpus of texts, previously unknown associations between terms and *domains* (i.e., disciplines, or fields of activity).[1]

The process generates, for each c_i in a set $C = \{c_1, \ldots, c_m\}$ of predefined domains, a lexicon L_1^i, bootstrapping from a lexicon L_0^i given as input. Associations between terms and domains are learned from a set θ of *unlabeled* (i.e., not tagged with domain labels) textual documents (hereafter called *corpus*). The process builds the lexicons $L_1 = \{L_1^1, \ldots, L_1^m\}$ for all the domains $C = \{c_1, \ldots, c_m\}$ in parallel from the same corpus θ. The only requirement on θ is that at least some of the terms in each of the lexicons in $L_0 = \{L_0^1, \ldots, L_0^m\}$ should occur in it (if none among the terms in a lexicon L_0^j occurs in θ, then no new term is added to L_0^j). Iterating this process would further allow the expansion of L_1 into increasingly larger lexicons L_2, L_3, ..., by simply using new corpora of unlabeled documents.

The method we propose is inspired by *text categorization*, the activity of automatically building, by means of machine learning techniques, automatic text classifiers, that is, programs capable of labeling natural language texts with (zero, one, or several) thematic categories from a predefined set $C = \{c_1, \ldots, c_m\}$ [Sebastiani 2002]. The construction of an automatic text classifier

[1]We want to point out that our use of the word "term" is somewhat different from the one often used in natural language processing and terminology extraction where it often denotes a *sequence* of lexical units expressing a concept of the domain of interest. Here we use this word in a neutral sense, that is, without making any commitment as to its consisting of a single word or a sequence of words.

requires the availability of a set $\psi = \{\langle d_1, C_1\rangle, \ldots, \langle d_h, C_h\rangle\}$ of preclassified documents, where a pair $\langle d_j, C_j\rangle$ indicates that document d_j belongs to all and only the categories in $C_j \subseteq C$. A general inductive process (called the *learner*) automatically builds a classifier for the set C by learning the characteristics of C from a *training* set $Tr = \{\langle d_1, C_1\rangle, \ldots, \langle d_g, C_g\rangle\} \subset \psi$ of documents. Once a classifier has been built, its effectiveness (i.e., its capability to take the right categorization decisions) may be tested by applying it to the *test set* $Te = \{\langle d_{g+1}, C_{g+1}\rangle, \ldots, \langle d_h, C_h\rangle\} = \psi - Tr$ and checking the degree of correspondence between the decisions of the automatic classifier and those encoded in the corpus. While the purpose of text categorization is that of classifying documents represented as vectors in a space of terms, the purpose of *term* categorization, as we formulate it, is (dually) that of classifying terms represented as vectors in a space of documents. In this task, terms are thus items that may belong, and must thus be assigned, to (zero, one, or several) domains belonging to a predefined set. In other words, starting from a set Γ_0^i of preclassified terms, a new set of terms Γ_1^i is classified, and the terms in Γ_1^i which are deemed to belong to c_i are added to L_0^i to yield L_1^i. The set Γ_0^i is composed of lexicon L_0^i, acting as the set of positive examples of c_i, plus a set of terms known not to belong to c_i, acting as the set of negative examples of c_i.

For input to the learning device and to the term classifiers that this will eventually build, we use bag-of-documents representations for terms, dual to the "bag of terms" representations commonly used in text categorization. As the learning device, we adopt ADABOOST.MH^{KR} [Sebastiani et al. 2000; Nardiello et al. 2003], a more efficient variant of the ADABOOST.MH^{R} algorithm proposed in Schapire and Singer [2000]. Both algorithms are an implementation of *boosting*, a method for supervised learning which has successfully been applied to many different domains and which has proven one of the best performers in text categorization applications so far [Schapire and Singer 2000; Sebastiani et al. 2000; Nardiello et al. 2003]. Boosting is based on the idea of relying on the collective judgment of a committee of classifiers that are trained sequentially; in training, the k-th classifier special emphasis is placed on the correct categorization of the training examples which have proven harder for (i.e., have been misclassified more frequently by) the previously trained classifiers.

This article is organized as follows. In Section 2, we describe how we represent terms by means of a bag-of-documents representation. Section 3 presents our approach to the expansion of domain-specific lexicons, discussing the operational methodology for employing the approach in practice (Section 3.1) and the experimental methodology we have followed for testing it (Section 3.2). Section 4 describes the results of our experiments in which we attempt to expand (in parallel) several domain-specific lexicons (42 in some experiments, 145 in others) by using a corpus of more than 800,000 documents. In Section 5, we review related work on the automated generation of lexical resources and spell out the differences between our approach and existing approaches. Section 6 concludes, pointing to avenues for improvement.

2. REPRESENTING TERMS IN A SPACE OF DOCUMENTS

2.1 Representing Documents

The approach to term representation that the IR community has almost universally adopted is a natural evolution of the approach that the very same community has developed for document representation. This latter approach (that we will here call the *term occurrence representation* (TOR)) assumes that a document d_j is represented as a vector of *term weights* $\vec{d}_j = \langle w_{1j}, \ldots, w_{rj} \rangle$, where r is the cardinality of the *dictionary* $\mathcal{T}$ and $0 \leq w_{kj} \leq 1$ represents, loosely speaking, the contribution of term t_k to the specification of the semantics of d_j. Usually, the dictionary is equated with the set of terms that occur at least once in at least α documents in the training set Tr (with α a predefined threshold, typically ranging between 1 and 5).

Different approaches to document representation may result from different choices (i) as to what a term is, and (ii) as to how term weights should be computed. A frequent choice for (i) is to use single words (minus "stop words," i.e., topic-neutral words such as articles and prepositions which are usually removed in advance) or their *stems* (i.e., their morphological roots). Different weighting functions may be used for tackling issue (ii); a frequent choice is the (cosine-normalized) *tfidf* function, where two intuitions are at play: (a) the more frequently t_k occurs in d_j, the more important for d_j it is (the *term frequency assumption*); (b) the more documents t_k occurs in, the smaller its contribution is in characterizing the semantics of a document in which it occurs (the *inverse document frequency assumption*). Weights computed by *tfidf* techniques are often normalized so as to contrast the tendency of *tfidf* to emphasize long documents. The version of *tfidf* that will provide the inspiration for the term representations discussed in this article is[2]

$$tfidf(t_k, d_j) = tf(t_k, d_j) \cdot \log \frac{|\mathcal{D}|}{\#_{\mathcal{D}}(t_k)}, \tag{1}$$

where $\#_{\mathcal{D}}(t_k)$ denotes the number of documents in the document collection $\mathcal{D}$ in which t_k occurs at least once and

$$tf(t_k, d_j) = \begin{cases} 1 + \log \#(t_k, d_j) & \text{if } \#(t_k, d_j) > 0 \\ 0 & \text{otherwise} \end{cases},$$

where $\#(t_k, d_j)$ denotes the number of times t_k occurs in d_j. In Equation (1), the $tf(t_k, d_j)$ factor is called *term frequency*, while the $\log \frac{|\mathcal{D}|}{\#_{\mathcal{D}}(t_k)}$ factor is called *inverse document frequency*. Weights obtained by Equation (1) are then normalized by means of cosine normalization, finally yielding

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|\mathcal{T}|} tfidf(t_s, d_j)^2}}. \tag{2}$$

[2]We stress that our use of this particular form of *tfidf* (and our use of *tfidf* itself, for that matter) is just as a proof of concept. The arguments we put forth in this article are independent of the weighting function used, and any other function could have been used for this purpose.

2.2 Representing Terms

The term representation that the IR community has almost universally adopted (that we will here call the *document occurrence representation* (DOR)) is a dual version of the document representation discussed in Section 2.1, and embodies the idea that, as the semantics of a document may be viewed as a function of the bag of terms that occur in it, the semantics of a term may be viewed as a function of the bag of documents in which the term occurs. A term t_j is then represented as a vector of *document weights* $\vec{t}_j = \langle w_{1j}, \ldots, w_{rj} \rangle$, where r is the cardinality of the document collection $\mathcal{D}$, and $0 \leq w_{kj} \leq 1$ represents the contribution of d_k to the specification of the semantics of t_j. The very same functions that were used for weighting the contribution of terms in document representations can be used for weighting the contribution of documents in term representations. *Mutatis mutandis*, the *tfidf* function of Equation (1), now aptly renamed the *dfitf* function, is reinterpreted as follows:

$$dfitf(d_k, t_j) = df(d_k, t_j) \cdot \log \frac{|\mathcal{T}|}{\#_{\mathcal{T}}(d_k)}, \tag{3}$$

where $\#_{\mathcal{T}}(d_k)$ denotes the number of distinct terms in the dictionary $\mathcal{T}$ which occur at least once in d_k and

$$df(d_k, t_j) = \begin{cases} 1 + \log \#(d_k, t_j) & \text{if } \#(d_k, t_j) > 0 \\ 0 & \text{otherwise} \end{cases},$$

where $\#(d_k, t_j)$ denotes the number of times t_k occurs in d_j. Weights obtained by Equation (3) are normalized by cosine normalization, finally yielding

$$w_{kj} = \frac{dfitf(d_k, t_j)}{\sqrt{\sum_{s=1}^{|\mathcal{D}|} dfitf(d_s, t_j)^2}}. \tag{4}$$

Symmetrically to the case discussed in Section 2.1, the intuitions are that (a) the more frequently t_i occurs in d_k, the more important d_k is for characterizing the semantics of t_i; (b) the more distinct terms d_k contains, the smaller its contribution is in characterizing the semantics of a term t_i which occurs in it.

Incidentally, it is interesting to note that, in switching from text indexing to term indexing, *idf* and cosine normalization switch their roles: the intuition that terms occurring in many documents should be deemphasized is implemented by *idf* in text indexing and cosine normalization in term indexing, while the intuition that longer documents need to be deemphasized is implemented by cosine normalization in text indexing and (the now more aptly renamed) *itf* in term indexing.

This approach to term representation is very elegant in that it is based on a minimal set of assumptions (namely, the "extensional" assumption that objects can be represented as bags of features, and the assumption that occurrence can be used as featurehood) and can be instantiated by means of any indexing technique (here we have used cosine-normalized *tfidf*) either from the tradition of text indexing or not. Note also that any program or data structure that

implements a text indexing function may be used straightaway for term indexing with no modification; one needs only to feed the program with the term identifiers in place of the document identifiers and vice versa. There is thus a clear symmetry between terms and documents in the sense that one may determine the meaning of the other, depending on one's viewpoint. The only aspect for which the symmetry breaks in practice is that, both in document and term indexing, we tend to directly pick a set of documents to work with, and the set of terms we work with is (indirectly) determined as a consequence in the sense that it coincides with the set of terms appearing in the chosen documents. Therefore, it is usually the case that documents are the independent variables and terms are the dependent variables of our problem, whatever the problem.

A straightforward consequence of this approach is that semantic relatedness between terms is viewed as a function of *term co-occurrence*, a heuristic notion which has been widely used in many past approaches to lexicon generation (see Section 5). In fact, according to our chosen approach, two terms have the highest similarity when they occur exactly in the same documents and with the same frequency, and are very similar when they co-occur in a high proportion of documents and with similar frequencies. However, while the past approaches discussed in Section 5 only deal with a "pure" version of co-occurrence, our term indexing approach brings about, for free, a weighted and length-normalized notion of co-occurrence, thus injecting higher sophistication into the measure of term similarity.

3. GENERATING DOMAIN-SPECIFIC LEXICONS BY SUPERVISED LEARNING

3.1 Operational Methodology

We are now ready to describe the overall process that we will follow for the expansion of domain-specific lexicons. We start from a set of domain-specific lexicons $L_0 = \{L_0^1, \ldots, L_0^m\}$, one for each domain in $C = \{c_1, \ldots, c_m\}$, and from a corpus θ. We index the terms that occur in L_0 by means of the term indexing technique described in Section 2.2. This yields, for each term t_k, a representation consisting of a vector of weighted documents where the length of the vector is $r = |\theta|$. By using the terms in $\{L_0^1, \ldots, L_0^m\}$ as positive training examples for $\{c_1, \ldots, c_m\}$, respectively, and by choosing negative training examples suitably (see Section 4.3), we then generate m classifiers $\Phi = \{\Phi^1, \ldots, \Phi^m\}$ by applying the AdaBoost.MH^{KR} algorithm. Note that the m classifiers are independent, which means that a given term may be classified into zero, one, or several domains at the same time.

Note that AdaBoost.MH^{KR}, like its predecessor AdaBoost.MH^{R}, uses binary vectorial representations. This will mean that, in the AdaBoost.MH^{KR} experiments, a binary version of the dual indexing approach of Section 2.2 will be used (which consists in assigning a weight of 1 or 0 to a document in which the term occurs or does not occur, respectively). The full-fledged version of the dual indexing approach will instead be used in the experiments of Section 4.4.7 in which we replace AdaBoost.MH^{KR} with a learner (SVMlight) that uses nonbinary input.

Table I.

Domain c_i		expert judgments	
		YES	**NO**
classifier judgments	**YES**	TP_i	FP_i
	NO	FN_i	TN_i

The contingency table for domain c_i. Here, FP_i (*false positives wrt* c_i) is the number of test terms incorrectly classified under c_i; TN_i (*true negatives wrt* c_i), TP_i (*true positives wrt* c_i) and FN_i (*false negatives wrt* c_i) are defined accordingly

3.2 Experimental Methodology

The process we have described in Section 3.1 is the one that we would apply in an operational setting. In an experimental setting, we are also interested in evaluating the effectiveness of our approach on a benchmark. The difference with the process outlined in Section 3.1 is that, at the beginning of the process, the lexicon L_0 is split into a training set and a test set; the classifiers are learned from the training set and are then tested on the test set by checking how good they are at extracting the terms in the test set from the corpus θ.

We will comply with standard text categorization practice in evaluating term categorization effectiveness by a combination of *precision* (π), the percentage of positive categorization decisions that turn out to be correct, and *recall* (ρ), the percentage of positive correct categorization decisions that are actually made. Since most classifiers can be tuned to emphasize one at the expense of the other, only combinations of the two are usually considered significant. Following common practice, as a measure combining the two, we will adopt their harmonic mean, that is, $F_1 = \frac{2\pi\rho}{\pi+\rho}$. Effectiveness will be computed with reference to the contingency table illustrated in Table I. When effectiveness is computed for several domains, the results for individual domains must be averaged in some way. We will do this both by *microaveraging* (domains count proportionally to the number of their positive test examples), that is,

$$\pi^{\mu} = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FP_i)} \tag{5}$$

$$\rho^{\mu} = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FN_i)} \tag{6}$$

and by *macroaveraging* (all domains count the same), that is,

$$\pi^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} \pi_i}{m} \qquad \rho^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} \rho_i}{m}. \tag{7}$$

Here, μ and M indicate microaveraging and macroaveraging, respectively, while the other symbols are as defined in Table I. Microaveraging rewards classifiers that behave well on *frequent domains* (i.e., domains with many positive test examples), while classifiers that perform well also on infrequent domains are emphasized by macroaveraging. Whether one or the other should be adopted obviously depends on the application requirements.

4. EXPERIMENTS

In order to test our approach according to the methodology described in Section 3.2, we need two types of resources: (i) a corpus θ of documents which provides the implicit representation for terms, and (ii) a set of domain-specific lexicons $L_0 = \{L_0^1, \ldots, L_0^m\}$.

We now describe the resources we used in our experiments.

4.1 The Corpus

As the corpus θ, we used the Reuters Corpus Volume 1 (RCV1)[3], a set of documents made available by Reuters for text categorization experimentation and consisting of the 806,812 news stories produced by Reuters from 20 Aug 1996 to 19 Aug 1997; all news stories are in English, and have 109 distinct terms per document on average [Rose et al. 2002]. Note that, although the texts of RCV1 are labeled by thematic categories, we did not make use of such labels (nor would it have made sense to use them, given that these categories are very different from and basically unrelated to the ones we are working with). The reason why we chose this corpus is that it provides a large enough set of data and that it is domain-generic, that is, it is not related in any semantically significant sense to the domains we used in the experiments.

4.2 The Lexicons

As the domain-specific lexicons, we used an extension of WordNet called WordNetDomains. WordNet [Fellbaum 1998] is a large, widely available, domain-generic, monolingual, machine-readable dictionary in which sets of synonymous words are grouped into 99,642 synonym sets (or *synsets*) organized in a directed acyclic graph. WordNet contains 173,941 word senses and 129,502 different lemmas; among these latter, 94,474 are nouns. In this work, we will refer to WordNet version 1.6.

In WordNet, only a few synsets are labeled with thematic categories, mainly contained in the glosses. This limitation is overcome in WordNetDomains, an extension of WordNet built by Magnini and Cavaglià [2000], in which each synset has been labeled with one or more from a set of 164 thematic categories, called *domains*[4]. The 164 domains of WordNetDomains are a subset of the categories belonging to the classification scheme of Dewey Decimal Classification (DDC); example domains are BASKETBALL, SPORT, and ZOOLOGY[5]. These 164 domains have been chosen by Magnini and Cavaglià from the much larger set of DDC categories since they are the most popular labels used in dictionaries for sense discrimination purposes. Domains have long been used in lexicography (where they are sometimes called *subject labels*) to mark technical usages of words (see Figure 2 for an example). Although they convey useful information

[3] `http://trec.nist.gov/data/reuters/reuters.html`.

[4] From the point of view of our term categorization task, the fact that more than one domain may be attached to the same synset means that ours is a *multilabel* categorization task [Sebastiani 2002, Section 2.2].

[5] WordNetDomains is publicly available at `http://tcc.itc.it/research/textec/topics/disambiguation/download.html`.

bats-man *n* (*pl* -men) **1** (CRICKET) player who bats: *He's a good batsman but no good as a bowler*. **2** (AVIATION) man who uses a pair of bats (like those used in table-tennis) to guide an aircraft as and after it touches down (e g on the deck of an aircraft-carrier).

Fig. 2. Two domains (CRICKET and AVIATION) used as word sense discrimination devices in a dictionary entry for batsman. The example is drawn from the Oxford Advanced Learner's Dictionary of Current English, 3rd edition.

for sense discrimination, they typically tag only a small portion of a dictionary. WordNetDomains instead extends the coverage of domain labels to an entire, existing lexical database, that is, WordNet. In this work, we will refer to WordNetDomains Version 1.0 - 070501. Note that WordNetDomains is organized in a tree structure with 5 top-level domains and a depth of at most 4 levels.

A domain may include synsets of different syntactic types. For instance, the MEDICINE domain groups together senses from Nouns, such as `doctor#1` (the first among several senses of the word "doctor") and `hospital#1`, and from Verbs, such as `operate#7`. A domain may include senses from different WordNet subhierarchies. For example, SPORT contains senses such as `athlete#1` which descends from `life_form#1`; `game_equipment#1` from `physical_object#1`; `sport#1` from `act#2`; and `playing_field#1` from `location#1`. Note that domains may group different senses of the same word into different thematic clusters with the side effect of reducing WordNet word polysemy.

The annotation methodology used in Magnini and Cavaglià [2000] for creating WordNetDomains was mainly manual and based on lexico-semantic criteria which take advantage of the already existing conceptual relations in WordNet.

In some of the experiments reported in this article, we used a coarser-grained variant of WordNetDomains, called WordNetDomains(42). This was obtained by Magnini and Cavaglià from WordNetDomains by considering only 42 fairly general domains (those at the second level in the WordNetDomains tree, see Table II) and tagging by a given domain c_i also the synsets that, in WordNetDomains, were tagged by the domains immediately related to c_i in a hierarchical sense (i.e., the parent domain of c_i and all the children domains of c_i). For instance, the domain SPORT is retained into WordNetDomains(42) and labels the synsets that it originally labeled in WordNetDomains plus the ones that in WordNetDomains were labeled by its children domains (e.g. VOLLEY, BASKETBALL, etc.) or by its parent domain (FREE-TIME) which are not retained in WordNetDomains(42). Since FREE-TIME has another child (PLAY) which is also retained in WordNetDomains(42), the synsets originally labeled by FREE-TIME will now be labeled also by PLAY and will thus have multiple labels. However, that a synset may belong to multiple domains is true in general, that is, these domains need not have any particular relation in the hierarchy.

This restriction to the 42 most significant domains is meant to bring about a good compromise between the conflicting needs of avoiding data sparseness and preventing the loss of relevant semantic information. These 42 domains belong to 5 groups, where the domains in a given group are all the children of the same WordNetDomains domain, which is, however, not retained into WordNetDomains(42). For example, one group is formed by SPORT and PLAY which are both children of FREE-TIME (not included in WordNetDomains(42)).

Table II.

Domains	Training Terms	Test Terms	Domains	Training Terms	Test Terms
ADMINISTRATION	1739	855	LAW	722	382
AGRICULTURE	128	61	LINGUISTICS	568	297
ALIMENTATION	1070	482	LITERATURE	323	180
ANTHROPOLOGY	538	254	MATHEMATICS	249	120
ARCHAEOLOGY	32	15	MEDICINE	1077	573
ARCHITECTURE	1578	730	MILITARY	661	320
ART	889	440	PEDAGOGY	269	149
ARTISANSHIP	49	31	PHILOSOPHY	83	64
ASTROLOGY	21	12	PHYSICS	893	393
ASTRONOMY	181	69	PLAY	514	230
BIOLOGY	3375	1433	POLITICS	569	333
BODY-CARE	72	30	PSYCHOLOGY	993	581
CHEMISTRY	834	397	PUBLISHING	208	101
COMMERCE	368	189	RELIGION	869	405
COMPUTER-SCIENCE	263	89	SEXUALITY	150	72
EARTH	2201	1088	SOCIOLOGY	408	209
ECONOMY	1446	700	SPORT	1002	455
ENGINEERING	372	164	TELECOMMUNICATION	285	130
FASHION	492	182	TOURISM	305	152
HISTORY	619	354	TRANSPORT	983	480
INDUSTRY	496	205	VETERINARY	25	17

The domains in WordNetDomains(42), each with the number of training and test terms occurring at least once in the full RCV1.

4.3 Structure of the Experiments

Figures 3 to 10 report several experiments run for different choices (i) of the subset of RCV1 chosen as the corpus θ, (ii) of the set L_0 of domain-specific lexicons, (iii) of the set of documents that act as features in the classification process, and (iv) of what counts as a feature in this process. We first describe the structure of a generic experiment and then go on to describe the sequence of different experiments we run.

In our experiments so far, we have considered only nouns, thereby discarding words tagged by other syntactic types. Nouns are more relevant from an applicative point of view (e.g. in query expansion) and are probably easier to classify within domains since they tend to be more domain-specific than, for example, verbs or adverbs. We plan to also consider words other than nouns in future experiments.

Before running the experiments, we lemmatize all the documents in the corpus θ and annotate the lemmas with part-of-speech tags, both by means of the TREE-TAGGER package [Schmid 1994]. We also use the WordNet morphological analyser in order to resolve ambiguities and lemmatization mistakes. During this phase, we also recognize multiwords (i.e., terms consisting of more than one word, such as `recording equipment`) contained in WordNet. The lemmatization phase allows us to discard all terms belonging to syntactic types other than nouns.

After this step, we perform a term filtering phase in which we discard:

—all "empty terms", that is, WordNetDomains(42) terms that, since they are not contained in any document of the corpus θ, are represented by a vector of

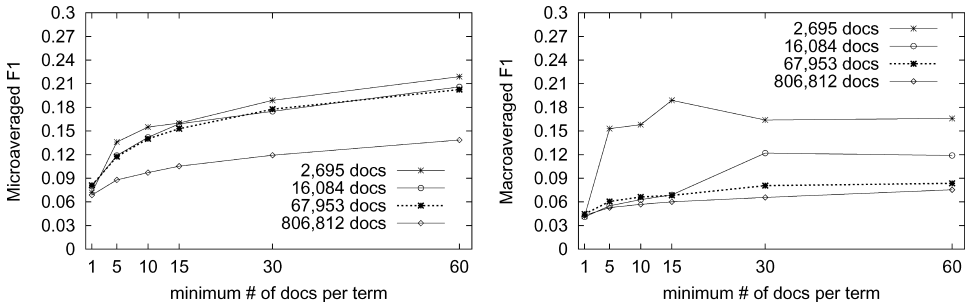

Fig. 3. Results obtained with the AdaBoost.MH^{KR} learner on subsets of RCV1 corresponding to one day's, one week's, one month's, and one year's worth of RCV1 data, and on the lexicons in WordNetDomains(42). Plots report micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) as a function of x which represents the minimal number of documents in which training and test terms must occur in order to be taken into consideration.

all zeroes. We discard them since (i) empty training terms could not possibly contribute to learning the classifiers, and (ii) empty test terms could not possibly be extracted by any algorithm that extracts terms from corpora;

—terms that occur in θ but do not belong to WordNetDomains(42) since they do not play any role in our experiments.

We repeat each term classification experiment several times by considering only training and test terms occurring in at least x documents for each value of $x \in \{1, 5, 10, 15, 30, 60\}$. Therefore, the curves describing our experiments all plot F_1 as a function of x; each curve in Figures 3 to 10 is the result of six different experiments (one for each value of $x \in \{1, 5, 10, 15, 30, 60\}$) since, for each different value of x, the experiment has to be repeated anew.

We randomly divide the set of the remaining terms into a training set Tr, consisting of two thirds of the entire set, and a test set Te, consisting of the remaining third. As negative training examples of category c_i, we choose all the training terms that are not positive examples of c_i. Finally, before learning the term classifiers, we perform a *document filtering* phase by discarding all documents that do not contain any term from Tr since they do not contribute to representing the meaning of training terms and thus could not possibly be of any help in building the classifiers.

Note that, in this entire process, we do not consider the grouping of terms into WordNet synsets; that is, the lexical units of interest in our application are the terms and not the synsets. The reason is that RCV1 is not a sense-tagged corpus and, since a given WordNet term may belong to more than one synset, for any term occurrence τ, it is not clear to which synset τ refers.

4.4 The Results

4.4.1 *Experiment 1 (The Basic Experiment).* In our first set of experiments (see the curves marked by black stars in Figure 3), we used only a subset of the RCV1 corpus (about 8% of its total size), corresponding to the news stories produced in an entire month (1 Nov 1996 to 30 Nov 1996, 67,953 documents) with the purpose of getting a feeling for the dimensions of the

Table III.

x	π^{μ}	ρ^{μ}	F_1^{μ}	π^{M}	ρ^{M}	F_1^{M}
1	0.705	0.043	0.081	0.717	0.023	0.045
5	0.735	0.064	0.118	0.793	0.032	0.061
10	0.737	0.077	0.140	0.774	0.035	0.066
15	0.755	0.085	0.153	0.812	0.036	0.068
30	0.763	0.101	0.178	0.843	0.042	0.081
60	0.757	0.117	0.203	0.836	0.044	0.084

Results (both micro- and macro-averaged) obtained on the automated lexicon generation task with the ADABOOST.MH^{KR} learner on a subset of RCV1 corresponding to one month's worth of RCV1 data.

problem that need investigation. For the same reason, in this set of experiments, we used only a small number of boosting iterations (500). There are 16,790 terms in WordNetDomains(42) after the term filtering phase described in Section 4.3.

The results in Figure 3 show a constant and definite improvement when higher values of x are used, despite the fact that, as we found, higher levels of x mean a higher degree of polysemy, that is, a higher average number of labels per term (e.g., this increases from 1.66 for $x = 1$ to 2.25 for $x = 60$) which tends to confuse a learning device. This behavior is not surprising since when a term occurs, for example, in one document only, this means that only one entry in the vector that represents the term is nonnull (i.e., significant); this means that the vector representation of this term is scarcely significant[6]. This is in sharp contrast to what happens in text categorization in which the number of nonnull entries in the vector representing a document equals the number of distinct terms contained in the document and is usually at least in the hundreds.

As shown in Table III, the low values obtained for F_1 are mostly the result of low recall values, while precision tends to be much higher. For instance, the F_1^{μ} value of .203 obtained for $x = 60$ is the result of the values $\pi^{\mu} = .760$ and $\rho^{\mu} = .117$. One way of improving F_1 could be tuning ADABOOST.MH^{KR} so as to increase recall at the expense of precision since F_1 is maximized when precision equals recall. Although this would be easy (e.g., by using the simple utility-theoretic technique described in Schapire et al. [1998] which consists in altering the initial distribution on which ADABOOST.MH^{KR} relies), we did not pursue this line of work for the simple reason that it would not bring interesting insights into the problem. That our F_1 values result from the combination of low recall and high precision values is true throughout the experiments described in the next pages. We conjecture that this is due to the following fact. In term (and text) categorization, unlike in many other machine learning applications, the number of negative examples of a given category c_i is usually overwhelmingly higher than the number of its positive examples (e.g., there

[6]By contrast, a fairly common term may have thousands of nonnull entries in the vector that represents it. This means that, in this application, the variance of the sparseness of the different vectors is tremendously high. This problem alone might deserve further investigation since this situation is rather unique within the field of vectorial representations.

are far fewer terms belonging to the domain AGRICULTURE than terms not belonging to it). If we use accuracy (the converse of error, i.e., $A = 1 - E$) as a measure of effectiveness, it will be very easy to generate an "effective" classifier since we simply need to generate the classifier that assigns every item d_j to $\overline{c}_i$ (*the trivial rejector*); in this way, the very few positive examples will have been misclassified, while the very many negative examples will have been correctly classified. This means that, when we use accuracy or error as the effectiveness measure that guides the learning process in boosting (or other learning mechanisms based on explicit effectiveness maximization) in this kind of application, we end up with a classifier that tends to behave like the trivial rejector, that is, it emphasizes precision at the expense of recall. Note, in fact, that the trivial rejector has maximum precision (since it never wrongly classifies a document under c_i) but minimum recall (since it never correctly classifies a document under c_i). In sum, relying, as we did, on a learning device based on explicit error minimization (as ADABOOST.MH^{KR} is) inevitably means generating classifiers with very high precision but very low recall.

In all the experiments that follow, we used the previously described set of experiments as a sort of baseline, testing whether modifying our approach along one or the other dimension of the problem could bring about a significant performance improvement. Note that, although the set of documents considered in the previous set of experiments is a subset of RCV1, it is a quite sizeable one, since it consists of 67,953 documents (more than 5 times the documents of the Reuters-21578 collection, a popular benchmark of text categorization research).

The following sections discuss the various dimensions of the problem that we explored.

4.4.2 *Experiment 2 (Using the Full RCV1).* In our next set of experiments, we run our system on the entire set of 806,812 RCV1 documents (this means 27,048 terms left in WordNetDomains(42) after term filtering) since we wanted to test whether performance can be improved by increasing significantly the number of documents from which terms have to be extracted. That this should happen might seem a plausible hypothesis since more documents mean, on average, a higher number of occurrences per term (on average, a training term occurs in 135.59 different documents in the "one month" set of experiments and 1,201.87 documents in the "one year" set of experiments), hence a more reliable indication of the typical contexts in which a given term occurs.

The results of this set of experiments (reported in Figure 3) indicate that this is not the case, since in going from 67,953 documents to 806,812 documents performance deteriorates. This trend resulting from the comparison of experiments run on one year's and one month's worth of documents is confirmed by two other sets of experiments we run using one week's (1 Nov 1996 to 7 Nov 1996, 16,003 documents) and one day's (1 Nov 1996, 2,689 documents) worth of documents (on average, a training term occurs in 46.45 different documents in the "one week" set of experiments and in 13.88 in the "one day" set of experiments). Altogether, the four sets of experiments clearly show that the fewer the documents, the better the performance.

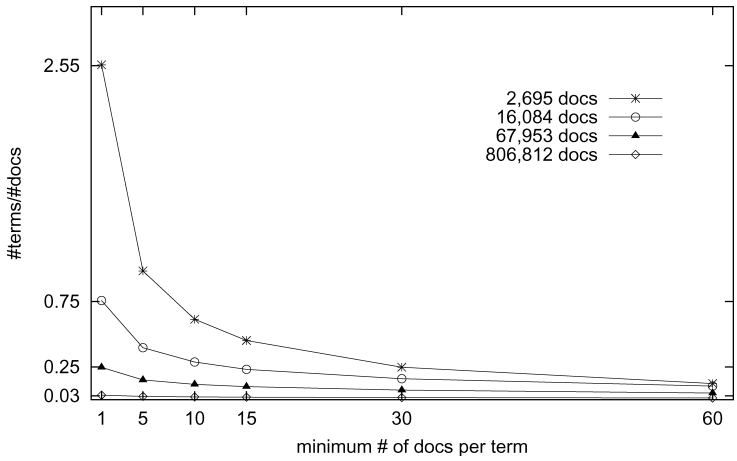

Fig. 4. Ratio between the number of training terms and the number of documents considered as a function of the x parameter.

A clear explanation of this fact is that increasing the number of documents produces a sharp decrease in the ratio between the number of training objects and the number of features that describe these objects (see Figure 4), a ratio that is conceptually akin to the one between the constraints of a problem and the number of its free variables. In other words, using the full RCV1 brings about an underconstrained problem, and the classifier tends to overfit the training data.

In future experiments, we plan to use even smaller numbers of features in order to determine the optimal number of features one should work with.

4.4.3 *Experiment 3 (Using Document Selection).* Our next set of experiments was aimed at verifying whether it might be the case that performance is depressed by RCV1 containing too many documents that are not significant enough in determining the meaning of our terms. These experiments consisted in first applying a pass of feature selection (in our case, document selection) aimed at choosing, from the 806,812 documents of RCV1, the ones that are the best discriminators between the presence and the absence of a domain, and then using the shortened term representations in which only the selected documents are retained as dimensions. Feature selection is often used by scoring each feature by means of a function that evaluates the capability of the feature at discriminating between the presence and the absence of a domain. In text categorization, Yang and Pedersen [1997] have shown that *information gain*, defined as

$$IG(t_k, c_i) = \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \cdot \log \frac{P(t, c)}{P(t) \cdot P(c)} \tag{8}$$

is one of the most effective such scoring functions. However, as specified in Equation (8), information gain evaluates the feature t_k with respect to a specific category c_i; in order to assess the value of a feature t_k in a global, category-independent sense, a globalization technique such as the maximum $IG_{\max}(t_k) = \max_{i=1}^{m} IG(t_k, c_i)$ of its category-specific values might be adopted.

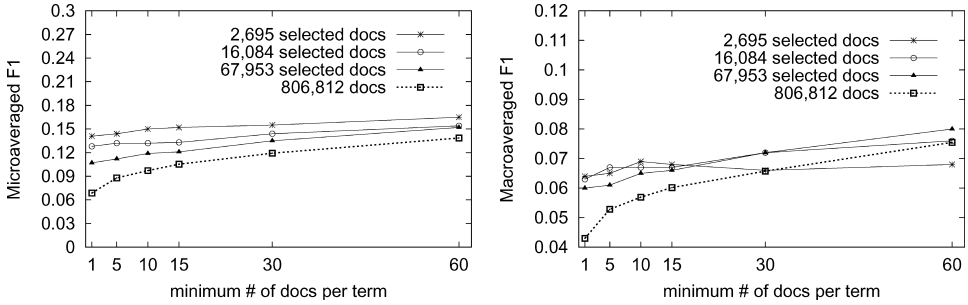

Fig. 5. Results obtained with the AdaBoost.MH^{KR} learner on the lexicons in WordNetDomains(42) and various amounts of documents selected from RCV1 via feature selection based on information gain.

In these experiments, we scored each RCV1 document by means of $IG_{\max}$ and retained only the top-scoring documents. In a first set of experiments, we retained the top 67,953 documents, corresponding to 8.72% of the original documents. This is exactly the same number of documents that were used in the "one month" set of experiments of Section 4.4.1 and was chosen in order to test what increase in performance (if any) could be achieved by replacing a set of k documents with a set of k "good" documents. For the same reason, our second set of experiments was run with the 16,084 top-scoring documents (corresponding to 2.06% of the original documents) and our third one with the 2,695 top-scoring documents (0.35% of the original documents), thus using exactly the same numbers of documents as used in the "one week" and "one day" experiments of Section 4.4.2.

The first observation (see Figure 5) is that each of these new sets of experiments produces an improvement with respect to the "one year" set of experiments of Section 4.4.2; the best performance was obtained with 0.35% of the original documents and the worst with 8.72%. However, at this point, it is not clear whether this benefit is due to feature selection or simply to the trend already noticed in Section 4.4.2, according to which smaller numbers of features tend to produce better results. In order to determine this, in Figure 6, we compare the performance obtained by using s "random" documents (as from the experiments of Sections 4.4.1 and 4.4.2) with that obtained by using s "good" documents (i.e., obtained through feature selection), where s ranges on the set {67,953; 16,084; 2,695}.

Here results are more difficult to explain since it seems that feature selection is really helpful for low values of the x parameter (e.g., for $x = 1$, performance is always much higher when feature selection was performed), while it is detrimental for high values of x (e.g., for $x = 60$ performance is always higher when feature selection was not performed). The improvement for the low values of x is easy to explain since classification theory tells us that s highly discriminating features are better than s random features when discrimination power is measured with respect to the target categories. The decrease in performance for the high values of x is more difficult to explain. We conjecture that this might be due to the fact that the experiments run in Sections 4.4.1 and 4.4.2

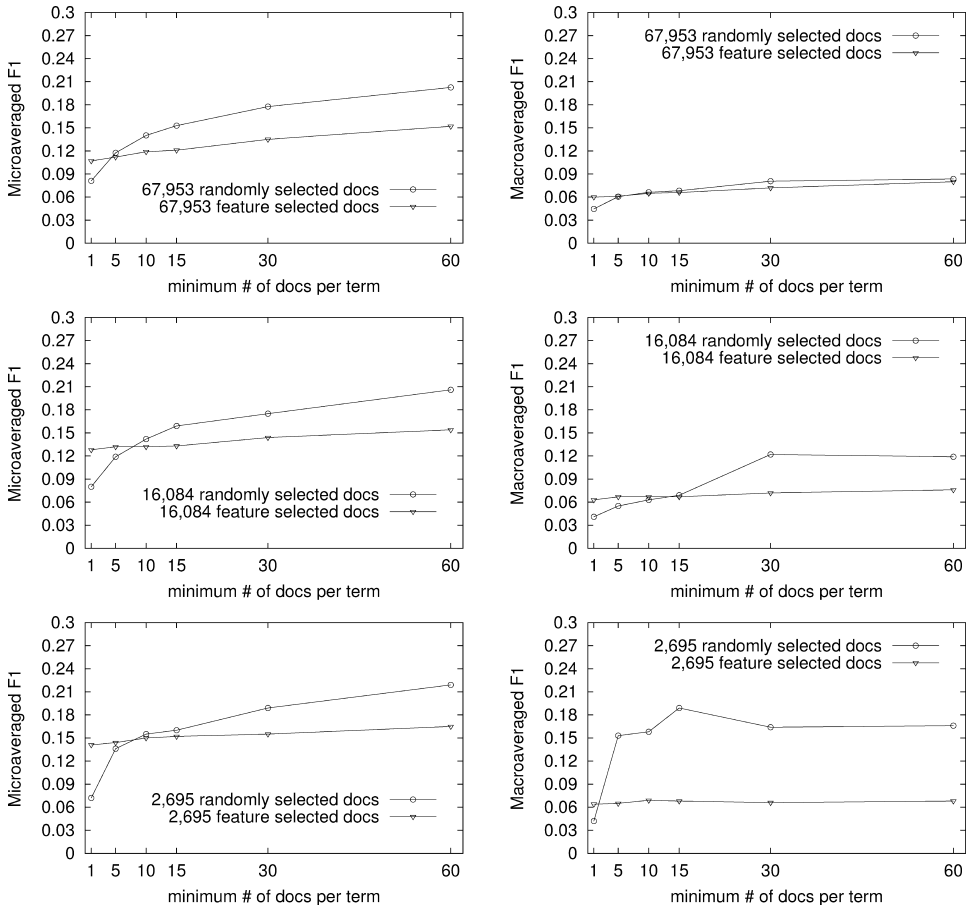

Fig. 6. Results obtained with the AdaBoost.MH^{KR} learner on the lexicons in WordNetDomains(42); each figure illustrates the difference between using s random documents and using s documents obtained via feature selection based on information gain ((top): s = 67,953; (mid): s = 16,084; (bottom): s = 2,695).

do not really use *s random* documents since these documents are temporally coherent, that is, they form a subinterval of the year being considered. Since the same subinterval typically contains several news stories about the same event, this means that, when a term occurs in several temporally coherent documents, there is a high chance that most if not all these occurrences pertain to the same sense (i.e., the degree of polysemy is reduced) and thus are more helpful in determining the domains of the words with which they co-occur. In order to clarify this aspect, in the future, we intend to run further experiments on the monosemic portion of WordNet.

4.4.4 *Experiment 4 (Using Sentences Instead of Documents).* In our next set of experiments, we reverted to the original set of 67,953 documents of Section 4.4.1 and tested whether sentences can be better features than full documents in term categorization. That this might be the case is plausible since,

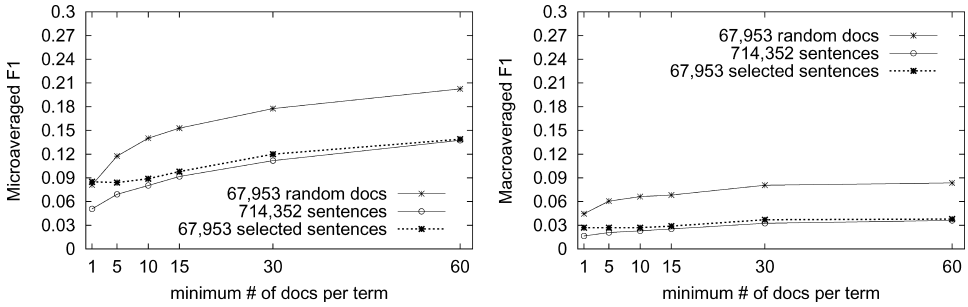

Fig. 7. Results obtained with the AdaBoost.MH^{KR} learner on the lexicons in WordNetDomains(42); each figure illustrates the difference between using 67,953 random documents, using all the sentences obtained by segmenting these 67,953 random documents, and using 67,953 sentences obtained from these latter via feature selection based on information gain.

given that (as observed in Section 2.2) our approach fundamentally relies on term co-occurrence, identifying the context in which co-occurrence is most significant is important. And it might be plausible to believe that the longer the context, the less significant the co-occurrence of two terms is in indicating that they belong to the same domain.

We run this set of experiments by segmenting each of the 67,953 documents into sentences (i.e., using the full stop as separator) and considering each of the resulting 714,352 sentences as a feature. (We did not run an experiment using paragraphs instead of sentences, that is, using a carriage return as the separator since most paragraphs in RCV1 documents consist of single sentences.) However, if we use all 714,352 sentences, we substantially decrease the ratio between the number of objects and the number of features that describe these objects, which means that, according to the observations of Section 4.4.2, our performance will likely decrease. In fact, this turns out to be the case, as shown in Figure 7.

As a result, we performed a further experiment in which we apply feature selection to the 714,352 sentences until we obtain a set of 67,953 top-scoring sentences. This is the same number of documents that was contained in the set of documents resulting from feature selection in the experiments of Section 4.4.3; we deliberately chose this number since we wanted to check if s good sentences are better than s "good" documents at discriminating domains.

The results (reported in Figure 7) seem to indicate that, for very aggressive feature selection, that is, when we retain only very few features, sentences are largely better features than documents, while this advantage tends to disappear for less aggressive levels. Note also that, in the former situation, performance is remarkably stable across all values of x, indicating that this setting is especially suitable for the situations in which one might want to extract very rare words.

This result seems consistent with an observation by Sahlgren [2004] who conjectures that co-occurrence within shorter linguistic contexts tends to be more indicative of true synonymy (or quasisynonymy), while co-occurrence in larger contexts tends to be more indicative of what he calls "topical relatedness" which is exactly our notion of a term being domain-specific.

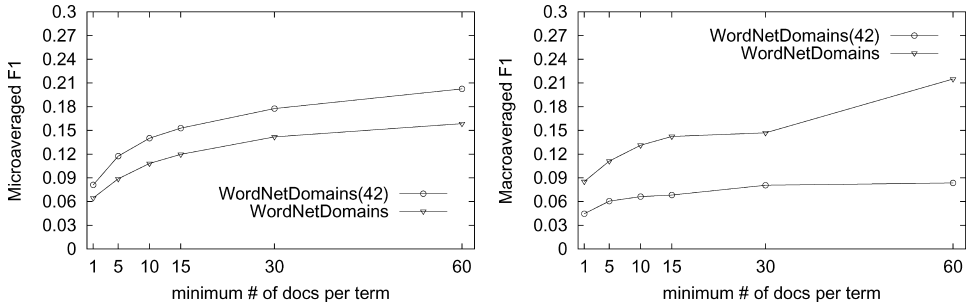

Fig. 8. Results obtained with the AdaBoost.MH^{KR} learner on one month's worth of RCV1 data and the lexicons either in WordNetDomains(42) or WordNetDomains with documents as textual units.

4.4.5 *Experiment 5 (Augmenting the Number of Boosting Iterations).* In our next set of experiments, we reverted to the full RCV1 with no feature selection and to using full documents instead of sentences and tested whether augmenting the number of AdaBoost.MH^{KR} iterations could improve the performance significantly. The results of this set of experiments have been fairly encouraging since, for $x = 1$, the value of F_1^{μ} increases from .068 (for 500 iterations) to .099 (1500 iterations) to .116 (2500 iterations). This improvement is due to a sharp increase in recall, while precision stays basically constant. We plan to explore this dimension of the problem more thoroughly in future experiments.

4.4.6 *Experiment 6 (Using the Full WordNetDomains).* In a further set of experiments, we reverted to the standard number of 500 iterations, and we used the full WordNetDomains instead of its subset WordNetDomains(42); this actually means using 145 domains, and not the full set of 164 domains that make up WordNetDomains, since 19 domains from WordNetDomains consist only of terms than never occur in RCV1 and thus have to be removed from consideration.

The aim of this set of experiments was testing whether the switch to more granular domains would improve or deteriorate the performance. An improvement of performance would be plausible on the grounds that the data these more granular domains contain are more significant since they are more focused. For instance, while the terms contained in the WordNetDomains domain Baseball are all focused on baseball, the terms contained in the WordNetDomains(42) domain Sport are more heterogeneous since they pertain to different sports. On the other hand, a deterioration in performance would also be plausible on the grounds that the more granular domains are more data-sparse, and less training data usually means worse performance.

The results were somehow inconclusive, as shown in Figure 8: by working on the full WordNetDomains, F_1^{μ} deteriorates while F_1^{M} improves. This clearly indicates an improvement in classification effectiveness over low-frequency categories and a deterioration in classification effectiveness over high-frequency categories.

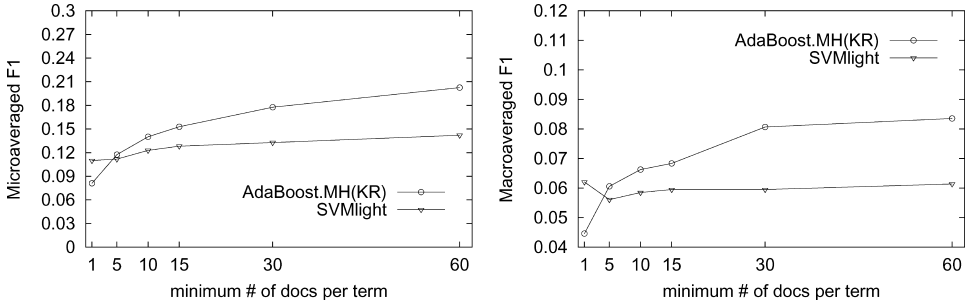

Fig. 9. Comparison of the results obtained on one month's worth of RCV1 data with the AdaBoost.MH^{KR} learner and with SVMlight.

4.4.7 *Experiment 7 (Switching to a Different Learner).* In order to check how dependent our results are on the choice of AdaBoost.MH^{KR} as learner, we performed a set of experiments in which AdaBoost.MH^{KR} is replaced by a support vector machine (SVM) learner implemented in the SVMlight package (version 3.5) [Joachims 1999], a well-known top performer of the text categorization field. SVM is a method that attempts to learn a hyperplane in $|\mathcal{T}|$-dimensional space that separates the positive training examples from the negative ones with the maximum possible margin, that is, such that the distance between the hyperplane and the training examples that are closest to it is maximum. Results in computational learning theory indicate that this tends to minimize the generalization error, that is, the error of the resulting classifier on yet unseen examples. We simply opted for the default parameter setting of SVMlight, both for efficiency reasons and in order to set a baseline for further work; in particular, this means that a linear kernel was used. As in the case of boosting, as negative training examples of domain c_i, we chose all the training terms that are not positive examples of c_i. For better comparability with AdaBoost.MH^{KR}, we run the same experiments as run with AdaBoost.MH^{KR} on one month's worth data (days from 01.11.1996 to 30.11.1996) with the same values of x as tested in the experiments of Figure 3.

As shown in Figure 9, SVMlight did not prove better than AdaBoost.MH^{KR}. Actually, the performance of SVMlight is almost uniformly worse than that of AdaBoost.MH^{KR}, particularly in the experiments with high values of x, and especially for macroaveraged effectiveness. However, we should note that the region on which SVMlight performs better than AdaBoost.MH^{KR} (namely, the low values of x) is an important one since these are the terms that occur quite infrequently, and, by virtue of this, they may be the most important ones to extract automatically since they are the ones that a lexicographer might easily miss when manually generating a lexicon.

This is yet another confirmation that the lexicon expansion task is a hard one since SVMs have been top performers in practically every machine learning application they have been used in, including text categorization.

As in the case of boosting, here too, our F_1 values are the result of low recall and high precision. Again, the fact that SVMs are inherently biased towards error minimization (see Section 4.4.1) seems the likely cause for this fact.

4.4.8 *Experiment 8 (Switching to a Different Representation).* In order to check how dependent our results are on the choice of the representation of terms discussed in Section 2.2, we performed a final set of experiments in which an alternative representation for terms was used. This representation, which we will call the *term co-occurrence representation* (in order to contrast it with the term occurrence representation of Section 2.1 and the document occurrence representation of Section 2.2), represents a term t_j by a vector $\vec{t}_j = \langle w_{1j}, \ldots, w_{rj} \rangle$ of weighted *terms*, where r is the cardinality of the dictionary (i.e., the set of terms that occur at least once in at least α documents of Tr, see Section 2.1 for comparison purposes), and the weight $0 \leq w_{kj} \leq 1$ represents, loosely speaking, the degree of semantic association between t_j and t_k as measured by the frequency with which they co-occur in the documents (note that it is always the case that $w_{jj} = 1$, but the practical effect of this fact is irrelevant). This representation is quite popular in the computational linguistics literature (see Dagan [2000] for a review) where it has been used for various purposes including word-sense disambiguation [Gale et al. 1993], the extraction of lexical collocations [Smadja 1993], and the extraction of syntactic relationships [Dagan et al. 1995].

Similar to the case of the term/text occurrence representations, several alternative weighting functions may be used. Here we rely again on a normalized *tfidf*-like weighting function, analogous to the one discussed in Section 2.2.

—The $tf(t_k, t_j)$ component of weight w_{kj} is now the number of documents in which t_k and t_j co-occur. This component emphasizes the weight of terms t_k that often co-occur with the term t_j we want to represent.

—The $idf(t_k)$ component of weight w_{kj} is the logarithm of the ratio between the size of the vocabulary and the number of different terms in the vocabulary with which t_k co-occurs at least once. This component de-emphasizes the weight of terms t_k that tend to occur with many other terms and whose co-occurrence with the term t_j we want to represent is thus less significant.

—The final weight w_{kj} is obtained by cosine-normalizing the results of the previous two steps, that is, $w_{kj} = \dfrac{tf(t_k, t_j) \cdot idf(t_k)}{\sqrt{\sum_{s=1}^{r} (tf(t_s, t_j) \cdot idf(t_s))^2}}$.

The results of these experiments, which we have run with both AdaBoost.MH^{KR} and SVMlight as learners, are reported in Figure 10. It can be seen that the term co-occurrence representation performs better. We think the reason may be due to the fact that this representation captures some phenomena related to semantic similarity better than the document occurrence representation. For instance, two perfect synonyms t_1 and t_2 may be represented by fairly dissimilar vectors according to the document occurrence representation since an author might typically use, in a given document, either the one or the other term, but not both, for better consistency. If all authors used this policy, t_1 and t_2 would never co-occur in the same document, hence yielding two highly different vectors. This need not be a problem with the term co-occurrence representation since the two terms need not frequently co-occur with each other in

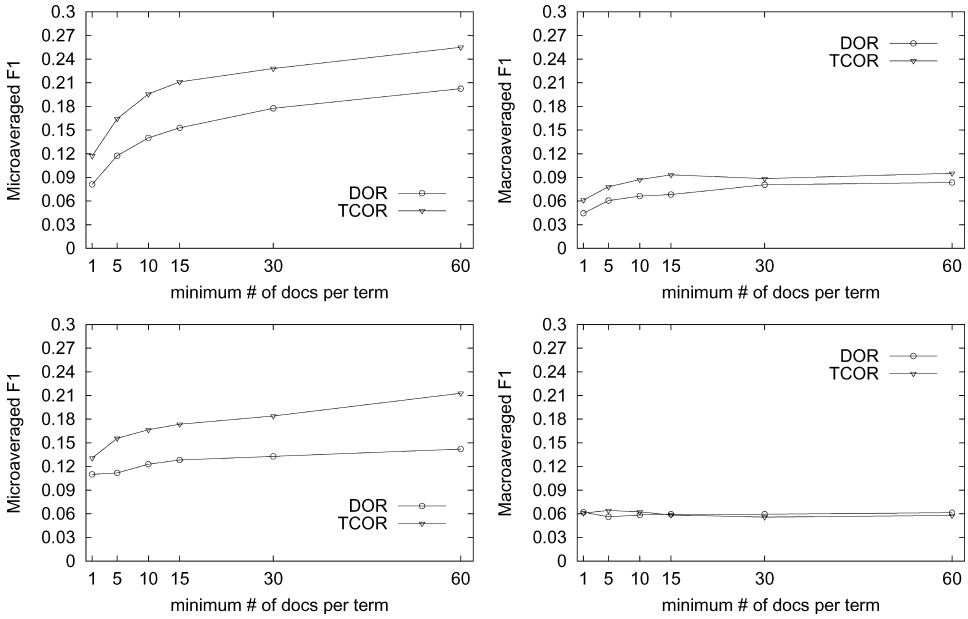

Fig. 10. Comparison of the results obtained with AdaBoost.MH^{KR} (top) and SVMlight (bottom) on one month's worth of RCV1 data with the standard document occurrence representation and the term co-occurrence representation.

order to be represented by highly similar vectors: they only need to frequently co-occur with the same terms, and this can plausibly happen given their perfect synonymy.

4.5 Comparison to Closely-Related Results

Note that in this article, we present no baseline, either published or new, against which to compare our results for the simple fact that there are no previous results in terms of both precision and recall on the term categorization task as we conceive it here.

To our knowledge, only Riloff and Shepherd [1999], Roark and Charniak [1998], and Thelen and Riloff [2002] have approached the problem of extending a domain-specific lexicon with new terms drawn from a text corpus. However, there are key differences between their evaluation methodology and ours which make comparisons problematic and unreliable.

First, their training terms (which they call *seed words*) have not been chosen randomly from a domain-specific dictionary but have been carefully selected through a manual process by the authors themselves. For instance, Riloff and Shepherd [1999] choose words that are "frequent in the domain" and that are "(relatively) unambiguous". Of course, their approach makes the task easier since it allows the best terms to be selected for training.

Second, Riloff and Shepherd [1999] and Roark and Charniak [1998] extract the terms from texts that are known to be strongly related to several among

the domains of interest[7] which makes the task easier than ours. Conversely, by using generic texts, (i) we can also expand domains for which no collection of texts is available, and (ii) we are able to expand domain-specific lexicons for (any set of) multiple domains in parallel from the same unlabeled text corpus, a task that, to the best of our knowledge, has never been investigated.

Third, the evaluation methodology of Riloff and Shepherd [1999] and Roark and Charniak [1998] is manual and *a posteriori*, in the sense that the authors themselves manually check the results of their own experiments, judging for each returned term how reasonable the inclusion of the term in the lexicon is. This contrasts with our evaluation methodology which is completely automatic (since we measure the proficiency of our system in discovering terms about the domain by the capability of the system to replicate the lexicon generation work previously performed by a lexicographer) and as such allows the experiments to be replicated by other researchers.

Fourth, checking one's results for reasonableness, as Riloff and Shepherd [1999] and Roark and Charniak [1998] do, means that one can only (a posteriori) measure precision (i.e., whether the terms spotted by the algorithm do in fact belong to the domain), but not recall (i.e., whether the terms belonging to the domain have actually been spotted by the algorithm). Again, this is in contrast with our methodology which (a priori) measures precision, recall, and a combination of them. Also, note that in terms of precision, the only measure that Riloff and Shepherd [1999] and Roark and Charniak [1998] (a posteriori) compute, our algorithm fares very well, mostly scoring higher than 70% (see Section 4.4.1)[8].

Aside from methodological issues, we should also note that our experiments are much larger in size that the ones presented in the quoted works. For instance, Thelen and Riloff [2002] work with about 1,700 texts, while we work with about 806,000; they report results on 6 domains, while we report on 42 or 145.

4.6 Why is Term Categorization Harder Than Text Categorization?

A fundamental observation that can be made from the results of our experiments is that the F_1 scores are much lower than the ones usually obtained in text categorization, even if using the same kind of "extensional representation + supervised learning" approach and the same top-performing learners. It is well known, for instance, that the very same SVM learner we have used in our experiments obtains microaveraged F_1 scores higher than .80 on the very same corpus (RCV1) we have used [Ault and Yang 2001; Lewis et al. 2004]. It is true that the set of classes used in these latter experiments is different from the ones

[7]The texts used here are the MUC-4 corpus which contains texts about terrorism. Categories of interest are, among others, MILITARY, TERRORIST, WEAPON.

[8]Thelen and Riloff [2002] also present recall figures for their experiments. However, their gold standard is not given by the set of all correctly classified terms but by the set of correctly classified terms extracted in a preprocessing phase by a term extraction algorithm. This set might thus not contain all terms, and might contain nonterms, which means that in principle their measure is not true recall. Also, they do not specify whether their recall figures are microaveraged or macroaveraged.

we have used in ours, but this fact alone cannot by itself justify such a large difference in performance. In other words, it is evident that term categorization is a harder task than text categorization.

Why this is so is not immediately clear, since the metaphor according to which the meaning of a text "coincides" with the terms it contains (a metaphor that has proven so successful in text categorization) is in principle no more powerful or intuitive than the dual metaphor according to which the meaning of a term "coincides" with the texts it is contained in (or with the similar metaphor according to which the meaning of a term "coincides" with the set of words it co-occurs with, as used in the experiments of Section 4.4.8).

We conjecture that the substantial difference in performance between the text and term categorization cases might be due to the fact that, while in text categorization there are often features with very high discriminative power, this does not seem to be the case for term categorization. To realize this, consider that in text categorization a perfect discriminator for category c_i is a term that occurs in all positive examples of c_i and never occurs in any negative examples of c_i. That such a perfect discriminator might exist in a real text categorization application is difficult but not impossible (when RCV1 is used as a text categorization collection, there are categories that have almost perfect discriminators). In our document occurrence representation for the term categorization task, a perfect discriminator for category c_i is a document in which all terms belonging to c_i occur and in which no term not belonging to c_i occurs. That such a perfect discriminator might exist seems not only difficult but plainly impossible since, when in natural language we want to state even the simplest fact about c_i, we also need terms which are not about c_i in order to express our thoughts. It is also evident that good discriminators for the term categorization task (i.e., documents that contain most terms belonging to c_i and contain only a few terms not belonging to c_i) will be extremely rare. Similar arguments can be made for the term co-occurrence representation we have used in the experiments of Section 4.4.8.

In order to have a more direct proof of this fact, we have computed, for our three extensional representations, the absolute values of the information gain function (globalized by means of the $f_{\max}$ method, as usual) for the features that score highest in terms of such a function. The results, which are computed on the entire RCV1 corpus of documents, are plotted in Figure 11. From the figure, we can see that on average a feature of the term occurrence representation (as used in text categorization) has a value of an order of magnitude higher than the feature of the same rank in the document occurrence representation (as used in term categorization). The term co-occurrence representation of Section 4.4.8 gives improved results with respect to the document occurrence representation, thus confirming the impression that it might be a better representation than document occurrence for term categorization applications. However, its results are of the same order of magnitude as the document occurrence representation.

Since information gain is a direct measure of the discriminative power of a feature, this comparison (although, as remarked earlier, not entirely fair because of the difference between the two category sets) is

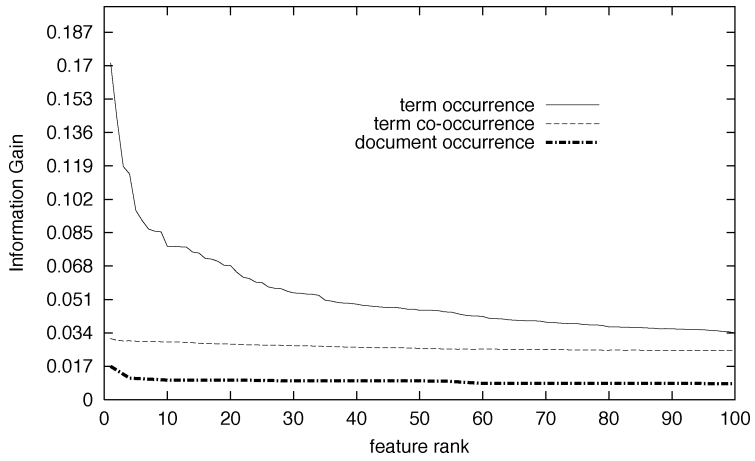

Fig. 11. Comparison of the absolute values of the information gain function for features in the term occurrence representation (as used in text categorization), in the document occurrence representation, and in the term co-occurrence representation (as used in term categorization) as a function of the rank of such features. The results are reported for the 100 top-scoring features only.

significant enough to indicate how much harder than text categorization term categorization is[9].

5. RELATED WORK

The automated generation of lexicons from text corpora has a long history, dating back at the very least to the seminal works of Lesk [1969], Salton [1971], and Spärck Jones [1971] and has been the subject of active research throughout the last 30 years both within the IR community [Crouch and Yang 1992; Jing and Croft 1994; Qiu and Frei 1993; Ruge 1992; Schütze and Pedersen 1997] and the NLP community [Grefenstette 1994; Hirschman et al. 1988; Riloff and Shepherd 1999; Roark and Charniak 1998; Tokunaga et al. 1995]. Most of the lexicons built by these works come in the form of *clustered thesauri*, that is, graphs in which nodes are groups of synonymous or quasisynonymous words, and edges connecting the nodes represent semantic contiguity. Most of these approaches follow the basic pattern of (i) measuring the degree of pairwise similarity between the words extracted from a corpus of texts, and (ii) clustering these words based on the computed similarity of values.

Step (i) requires terms to be given explicit (i.e., vectorial) representations so that similarity between them can be computed. For this, the two term representations we have tested in the experiments of Section 4.4.8, one in which the documents that contain the terms are the features [Crouch 1990; Crouch and Yang 1992; Qiu and Frei 1993; Schäuble and Knaus 1992; Sheridan and Ballerini 1996; Sheridan et al. 1997] and the other in which the co-occurring

[9]Note also that the performance of our experiments may also have been negatively influenced by the imperfect quality of the WordNetDomains resource which was generated by a combination of automatic and manual procedures and did not undergo extensive manual checking afterwards [Magnini and Cavaglià 2000].

terms are the features [Schütze 1992; Schütze and Pedersen 1997], can be alternatively chosen. The first representation is based on first-order co-occurrence (i.e., two terms are considered similar when they frequently co-occur with each other), while the latter is based on second-order co-occurrence (i.e., two terms are considered similar when they frequently co-occur with the same terms). Variants of this latter approach are obtained by restricting the context of co-occurrence from the document to the paragraph, or to the sentence, or to a sliding, fixed-size window of text centred around the focus term [Lund and Burgess 1996]. Other authors reinterpret the notion of co-occurrence as meaning something different from the mere simultaneous presence of the two terms in the same text window. For instance, Lin [1998] and Pantel and Lin [2003] represent term t_j by vectors of *pairs* (t_k, r_k), where t_k is a term that co-occurs with t_j in some sentence, and r_k is the grammatical relationship between t_j and t_k in this sentence; in this way, syntactic knowledge is brought to bear in what is otherwise an essentially knowledge-free approach.

When the lexical resources being built are domain-specific, whether a word belongs or not to the domain of interest is usually established by checking whether its frequency within documents belonging to the domain is higher than its frequency within generic documents [Chen et al. 1996; Riloff and Shepherd 1999; Schatz et al. 1996] (this property is often called *salience* [Yarowsky 1992]). This literature has thus taken an approach which can be summarized in the recipe "from a set of documents about domain c_i and a set of generic documents (i.e., mostly not about c_i) extract the words that mostly characterize c_i (and organize them into a thesaurus)". Our work is different, in that its underlying supervised approach requires a starting kernel of terms about c_i but does not require that the documents from which the terms are extracted be labeled according to the domains under consideration. This makes our technique particularly suitable for extending a previously existing domain-specific lexical resource, while the previously known unsupervised techniques tend to be more useful for generating one from scratch. This suggests an interesting methodology of (i) generating a domain-specific lexical resource by some unsupervised technique, and then (ii) extending it by our supervised technique.

As anyone involved in applications of supervised machine learning knows, labeled resources are often a bottleneck for learning algorithms since labeling items by hand is expensive. Concerning this, note that our technique is advantageous since it requires an initial set of labeled terms *only in the first iteration of the expansion process*, that is, for generating L_1 from L_0. Once a lexical resource has been extended with new terms, extending it further only requires a new *unlabeled* corpus of documents but no other labeled resource. This is different from the techniques described previously, which require, for extending a lexical resource that has just been built by means of them, a new labeled corpus of documents.

One advantage of our approach is that it solidly rests on the strong theoretical bases of indexing theory and supervised learning theory. It is also a minimalist approach in the sense that nothing other than an indexing tool and a supervised learning tool are needed for it; indexing tools and supervised learning tools different from the ones used here can be plugged in and out in a straightforward

manner. The approaches adopted in Riloff and Shepherd [1999], Roark and Charniak [1998], and Thelen and Riloff [2002], which have been discussed in more detail in Section 4.5, are also supervised (in the sense that they require the presence of training terms) but rest on heuristic pattern extraction techniques and scoring functions rather than on general supervised learning techniques. Also, the approaches presented in the quoted works have been tested only on texts which are at least loosely connected with several among the domains of interest; it remains to be seen how their techniques would work on corpora unrelated to the domains.

6. CONCLUSION AND DISCUSSION

We have reported an approach to the automatic expansion of domain-specific lexicons by the combination of (i) a dual interpretation of IR-style text indexing theory and (ii) a supervised learning approach. This approach allows the extraction of domain-specific terms from domain-generic texts. The advantages of our method are that it is particularly suited (i) to the situation in which several domain-specific lexicons need to be extended in parallel and (ii) to the case in which no labeled text corpora are available, and that it does not require preexisting semantic knowledge.

We have exemplified our approach by running experiments in which we simultaneously expand a large number of domain-specific lexicons (up to 145) by extracting new terms from a large number of domain-generic texts (up to 806,812). These experiments have been run by using widely available resources (the WordNetDomains set of domain-specific lexicons and the RCV1 corpus) and standard evaluation measures (precision and recall); this means that our results constitute a useful experimental setting and a baseline for other researchers to improve upon. We also hope that this study will encourage researchers to adopt for the lexicon expansion task the a priori evaluation methodology exemplified here.

Our experiments suggest that our approach is viable, although large margins of improvement still exist: F_1 values are still low, at least if compared to the F_1 values that have been obtained in text categorization research on the same corpus, so work is still needed in tuning this approach in order to obtain significant categorization performance. One obvious direction is to combine all the parameter settings that we have found to work best in individual experiments that explored individual facets of the problem (e.g., using a small number of texts, a fine-grained set of domains, a high number of boosting iterations, etc., at the same time), but many other directions exist. Even so, we have provided a theoretical argument which explains why we cannot hope to obtain for this term categorization task effectiveness values comparable to the ones which are routinely obtained in text categorization experiments.

To date, we have run experiments consisting of only one iteration of the expansion process. In future experiments, we also plan to allow for multiple iterations in which the system learns new terms from previously learned ones (similarly to the approach adopted in Riloff and Shepherd [1999] and Thelen and

Riloff[2002]). This is quite reasonable since our system displays a high precision (usually above .70) which tends to guarantee that it is safe for our system to learn from terms it has itself learned in the previous iterations. The very fact that our system has proven to perform best when fed with small quantities of text (e.g., a day's worth of newswire stories) thus suggests a simple strategy of performing many iterations of the process, each time using a small quantity of text.

ACKNOWLEDGMENTS

We thank Pio Nardiello for his assistance in tuning the ADABOOST.MH^{KR} code, Thorsten Joachims for making the SVMLIGHT package available, Alessandro Sperduti for valuable comments, and the anonymous reviewers for their helpful and constructive comments.

REFERENCES

AONE, C. AND BENNETT, S. W. 1996. Applying machine learning to anaphora resolution. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, S. Wermter, E. Riloff, and G. Scheler, Eds. Springer Verlag, Heidelberg, Germany, 302–314. (Lecture Notes in Computer Science, vol. 1040).

AULT, T. AND YANG, Y. 2001. kNN, Rocchio and metrics for information filtering at TREC-10. In *Proceedings of 10th Text Retrieval Conference (TREC-10)*. E. M. Voorhees, Ed. National Institute of Standards and Technology, Gaithersburg, MD. 84–93.

BRILL, E. AND RESNIK, P. 1994. A transformation-based approach to prepositional phrase attachment disambiguation. In *Proceedings of 15th International Conference on Computational Linguistics (COLING'94)*. Kyoto, Japan, 1198–1204.

CHEN, H., SCHUFFELS, C., AND ORWING, R. 1996. Internet categorization and search: A machine learning approach. *J. Visual Comm. Image Represent.* Special Issue on Digital Libraries, *7*, 1, 88–102.

CROUCH, C. J. 1990. An approach to the automatic construction of global thesauri. *Inform. Process. Manage. 26*, 5, 629–640.

CROUCH, C. J. AND YANG, B. 1992. Experiments in automated statistical thesaurus construction. In *Proceedings of 15th ACM International Conference on Research and Development in Information Retrieval (SIGIR'92)*. Kobenhavn, Denmark, 77–87.

DAGAN, I. 2000. Contextual word similarity. In *Handbook of Natural Language Processing*, R. Dale, H. Moisl, and H. Somers, Eds. Marcel Dekker Inc, New York, NY. Chapter 19, 459–476.

DAGAN, I., MARCUS, S., AND MARKOVITCH, S. 1995. Contextual word similarity and estimation from sparse data. *Comput. Speech Lang. 9*, 2, 123–152.

FELLBAUM, C., Ed. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA.

GALE, W., CHURCH, K., AND YAROWSKY, D. 1993. A method for disambiguating word senses in a large corpus. *Comput. Humanities 26*, 5/6, 415–439.

GREFENSTETTE, G. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

HIRSCHMAN, L., GRISHMAN, R., AND SAGER, N. 1988. Grammatically-based automatic word class formation. *Inform. Process. Manage. 11*, 1/2, 39–57.

HIRSCHMAN, L., LIGHT, M., BRECK, E., AND BURGER, J. D. 1999. DEEP READ: A reading comprehension system. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*. Gaithersburg, MD. 325–332.

JING, Y. AND CROFT, W. B. 1994. An association thesaurus for information retrieval. In *Proceedings of 4th International Conference Recherche d'Information Assistee par Ordinateur (RIAO'94)*. New York, NY. 146–160.

JOACHIMS, T. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. J. Burges, and A. J. Smola, Eds. The MIT Press, Cambridge, MA. Chapter 11, 169–184.

LESK, M. E. 1969. Word-word association in document retrieval systems. *Ameri. Document. 20*, 1, 27–38.

LEWIS, D. D., LI, F., ROSE, T., AND YANG, Y. 2004. Reuters Corpus Volume 1 as a text categorization test collection. *J. Machine Learn. Resea. 5*, 361–397.

LIN, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*. Montreal, Canada, 768–774.

LUND, K. AND BURGESS, C. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behav. Resear. Meth. Instrument. Comput. 28*, 2, 203–208.

MAGNINI, B. AND CAVAGLIÀ, G. 2000. Integrating subject field codes into WordNet. In *Proceedings of 2nd International Conference on Language Resources and Evaluation (LREC'00)*. Athens, Greece. 1413–1418.

MAGNINI, B., STRAPPARAVA, C., PEZZULO, G., AND GLIOZZO, A. 2002. The role of domain information in word sense disambiguation. *Natural Lang. Engin. 8*, 4, 359–373.

MOLDOVAN, D., HARABAGIU, S., PAŞCA, M., MIHALCEA, R., GOODRUM, R., GÎRJU, R., AND RUS, V. 1999. LASSO: A tool for surfing the answer net. In *Proceedings of 8th Text Retrieval Conference (TREC-8)*. Gaithersburg, MD. 175–183.

NARDIELLO, P., SEBASTIANI, F., AND SPERDUTI, A. 2003. Discretizing continuous attributes in AdaBoost for text categorization. In *Proceedings of 25th European Conference on Information Retrieval (ECIR'03)*, Pisa, Italy, Springer Verlag, 320–334.

PANTEL, P. AND LIN, D. 2003. Automatically discovering word senses. In *Proceedings of 3rd International Conference on Human Language Technology (HLT'03)*. Edmonton, CA, 21–22.

QIU, Y. AND FREI, H.-P. 1993. Concept-based query expansion. In *Proceedings of 16th ACM International Conference on Research and Development in Information Retrieval (SIGIR'93)*. Pittsburgh, PA. 160–169.

RILOFF, E. AND SHEPHERD, J. 1999. A corpus-based bootstrapping algorithm for semi-automated semantic lexicon construction. *J. Natural Lang. Engin. 5*, 2, 147–156.

ROARK, B. AND CHARNIAK, E. 1998. Noun phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*. Montreal, Canada, 1110–1116.

ROSE, T., STEVENSON, M., AND WHITEHEAD, M. 2002. The Reuters Corpus Volume 1—from yesterday's news to tomorrow's language resources. In *Proceedings of 3rd International Conference on Language Resources and Evaluation (LREC'02)*. Las Palmas, Spain, 827–832.

RUGE, G. 1992. Experiments on linguistically-based terms associations. *Inform. Process. Manage. 28*, 3, 317–332.

SAHLGREN, M. 2004. Random indexing of words in narrow context windows for vector-based semantic analysis. In *Acquisition and Representation of Word Meaning: Theoretical and Computational Perspectives*, A. Lenci, S. Montemagni, and V. Pirrelli, Eds. Istituti Editoriali Poligrafici Internazionali, Pisa, Italy.

SALTON, G. 1971. Experiments in automatic thesaurus construction for information retrieval. In *Proceedings of the IFIP Congress*. Vol. TA-2. Ljubljana, Yugoslavia, 43–49.

SCHAPIRE, R. E. AND SINGER, Y. 2000. BOOSTEXTER: a boosting-based system for text categorization. *Machine Learn. 39*, 2/3, 135–168.

SCHAPIRE, R. E., SINGER, Y., AND SINGHAL, A. 1998. Boosting and Rocchio applied to text filtering. In *Proceedings of 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR'98)*. Melbourne, Australia, W. B. Croft, A. Moffat, C. J. V. Rijsbergen, R. Wilkinson, and J. Zobel, Eds. ACM Press, New York, NY, Melbourne, AU, 215–223.

SCHATZ, B. R., JOHNSON, E. H., COCHRANE, P. A., AND CHEN, H. 1996. Interactive term suggestion for users of digital libraries: Using subject thesauri and co-occurrence lists for information retrieval. In *Proceedings of 1st ACM Digital Library Conference (DL'96)*. Bethesda, MD. 126–133.

SCHÄUBLE, P. AND KNAUS, D. 1992. The various roles of information structures. In *Proceedings of the 16th Annual Conference of the Gesellschaft für Klassifikation*, Dortmund, Germany, O. Opitz, B. Lausen, and R. Klar, Eds. 282–290. Springer Verlag, Heidelberg, Germany, 1993.

SCHMID, H. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK, 44–49.

SCHÜTZE, H. 1992. Dimensions of meaning. In *Proceedings of Supercomputing'92*. Minneapolis, MN, 787–796.

SCHÜTZE, H. AND PEDERSEN, J. O. 1997. A cooccurrence-based thesaurus and two applications to information retrieval. *Inform. Process. Manage. 33*, 3, 307–318.

SEBASTIANI, F. 2002. Machine learning in automated text categorization. *ACM Comput. Surv. 34*, 1, 1–47.

SEBASTIANI, F., SPERDUTI, A., AND VALDAMBRINI, N. 2000. An improved boosting algorithm and its application to automated text categorization. In *Proceedings of 9th ACM International Conference on Information and Knowledge Management (CIKM'00)*, McLean, VA. A. Agah, J. Callan, and E. Rundensteiner, Eds. ACM Press, New York, NY, 78–85.

SHERIDAN, P. AND BALLERINI, J.-P. 1996. Experiments in multilingual information retrieval using the SPIDER system. In *Proceedings of 19th ACM International Conference on Research and Development in Information Retrieval (SIGIR'96)*. Zürich, Switzerland. 58–65.

SHERIDAN, P., BRASCHLER, M., AND SCHÄUBLE, P. 1997. Cross-language information retrieval in a multi-lingual legal domain. In *Proceedings of 1st European Conference on Research and Advanced Technology for Digital Libraries (ECDL'97)*, Italy, C. Peters and C. Thanos, Eds. Pisa, IT, 253–268. Lecture Notes in Computer Science, vol. 1324, Springer Verlag, Heidelberg, Germany.

SMADJA, F. 1993. Retrieving collocations from text: XTRACT. *Computation. Linguist. 19*, 1, 143–178.

SODERLAND, S., FISHER, D., ASELTINE, J., AND LEHNERT, W. 1995. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*. Montreal, Canada, 1314–1319.

SPÄRCK JONES, K. 1971. *Automatic Keyword Classification for Information Retrieval*. Butterworths, London, UK.

THELEN, M. AND RILOFF, E. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of 7th Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*. Philadelphia, PA, 214–221.

TOKUNAGA, T., IWAYAMA, M., AND TANAKA, H. 1995. Automatic thesaurus construction based on grammatical relations. In *Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*. Montreal, Canada, 1308–1313.

YANG, Y. AND PEDERSEN, J. O. 1997. A comparative study on feature selection in text categorization. In *Proceedings of 14th International Conference on Machine Learning (ICML'97)*, Nashville, TN, D. H. Fisher, Ed. Morgan Kaufmann Publishers, San Francisco, US, 412–420.

YAROWSKY, D. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of 14th International Conference on Computational Linguistics (COLING'92)*. Nantes, France, 454–460.

Received December 2005; accepted March 2006 by Kishore Papineni