Building Thematic Lexical Resources by Bootstrapping and Machine Learning

Alberto Lavelli[*], **Bernardo Magnini**[*], **Fabrizio Sebastiani**[†]

[*]ITC-irst
Via Sommarive, 18 – Località Povo
38050 Trento, Italy
{lavelli,magnini}@itc.it

[†]Istituto di Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
fabrizio@iei.pi.cnr.it

Abstract

We discuss work in progress in the semi-automatic generation of *thematic lexicons* by means of *term categorization*, a novel task employing techniques from information retrieval (IR) and machine learning (ML). Specifically, we view the generation of such lexicons as an iterative process of learning previously unknown associations between terms and *themes* (i.e. disciplines, or fields of activity). The process is iterative, in that it generates, for each c_i in a set $C = \{c_1, \ldots, c_m\}$ of themes, a sequence $L_0^i \subseteq L_1^i \subseteq \ldots \subseteq L_n^i$ of lexicons, bootstrapping from an initial lexicon L_0^i and a set of text corpora $\Theta = \{\theta_0, \ldots, \theta_{n-1}\}$ given as input. The method is inspired by *text categorization*, the discipline concerned with labelling natural language texts with labels from a predefined set of themes, or categories. However, while text categorization deals with documents represented as vectors in a space of terms, we formulate the task of term categorization as one in which terms are (dually) represented as vectors in a space of documents, and in which terms (instead of documents) are labelled with themes. As a learning device, we adopt *boosting*, since (a) it has demonstrated state-of-the-art effectiveness in a variety of text categorization applications, and (b) it naturally allows for a form of "data cleaning", thereby making the process of generating a thematic lexicon an iteration of generate-and-test steps.

1. Introduction

The generation of *thematic lexicons* (i.e. lexicons consisting of specialized terms, all pertaining to a given theme 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 to support text search and other information retrieval applications in the context of thematic, "vertical" portals (aka *vortals*)[1]. Vortals are a recent phenomenon in the World Wide Web, and have grown out of the users' needs for directories, services and information resources that are both rich in information and specific to their interests. This has led to Web sites that specialize in aggregating market-specific, "vertical" content and information. Actually, the evolution from the generic portals of the previous generation (such as Yahoo!) to today's vertical portals is just natural, and is no different from the evolution that the publishing industry has witnessed decades ago with the creation of specialized magazines, targeting specific categories of readers with specific needs. To read about the newest developments in ski construction technology, skiers read specialty magazines about skiing, and not generic newspapers, and skiing magazines is also where advertisers striving to target skiers place their ads in order to be the most effective. Vertical portals are the future of commerce and information seeking on the Internet, and supporting sophisticated information access capabilities by means of thematic lexical resources is thus of the utmost importance.

Unfortunately, the generation of thematic lexicons is expensive, since it requires the intervention of specialized manpower, i.e. 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 of cheaper and faster methods for answering application needs than manual lexicon generation. Also, as noted in (Riloff and Shepherd, 1999), the manual approach is prone to errors of omission, in that a lexicographer may easily overlook infrequent, non-obvious terms that are nonetheless important for many tasks.

Many applications also require that the lexicons be not only thematic, but also tailored to the specific data tackled in the application. For instance, in query expansion (automatic (Peat and Willett, 1991) or interactive (Sebastiani, 1999)) for information retrieval systems addressing thematic document collections, terms synonymous or quasi-synonymous to the query terms are added to the query in order to retrieve more documents. In this case, the added terms should occur in the document collection, otherwise they are useless, and the relevant terms which occur in the document collection should potentially be added. That is, for this application the ideal thematic lexicon should contain all and only the technical terms present in the document

[1]See e.g. `http://www.verticalportals.com/`

collection under consideration, and should thus be generated directly from this latter.

1.1. Our proposal

In this paper we propose a methodology for the semi-automatic generation of thematic lexicons from a corpus of texts. This methodology relies on *term categorization*, a novel task that employs a combination of techniques from information retrieval (IR) and machine learning (ML). Specifically, we view the generation of such lexicons as an iterative process of learning previously unknown associations between terms and *themes* (i.e. disciplines, or fields of activity)[2]. The process is iterative, in that it generates, for each c_i in a set $C = \{c_1, \ldots, c_m\}$ of predefined themes, a sequence $L_0^i \subseteq L_1^i \subseteq \ldots \subseteq L_n^i$ of lexicons, bootstrapping from a lexicon L_0^i given as input. Associations between terms and themes are learnt from a sequence $\Theta = \{\theta_0, \ldots, \theta_{n-1}\}$ of sets of documents (hereafter called *corpora*); this allows to enlarge the lexicon as new corpora from which to learn become available. At iteration y, the process builds the lexicons $L_{y+1} = \{L_{y+1}^1, \ldots, L_{y+1}^m\}$ for all the themes $C = \{c_1, \ldots, c_m\}$ in parallel, from the same corpus θ_y. The only requirement on θ_y is that at least some of the terms in each of the lexicons in $L_y = \{L_y^1, \ldots, L_y^m\}$ should occur in it (if none among the terms in a lexicon L_y^j occurs in θ_y, then no new term is added to L_y^j in iteration y).

The method we propose is inspired by *text categorization*, the activity of automatically building, by means of machine learning techniques, *automatic text classifiers*, i.e. programs capable of labelling 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 requires the availability of a corpus $\psi = \{\langle d_1, C_1\rangle, \ldots, \langle d_h, C_h\rangle\}$ 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) themes belonging to a predefined set. In other words, starting from a set Γ_y^i of preclassified terms, a new set of terms Γ_{y+1}^i is classified, and the terms in Γ_{y+1}^i which are deemed to belong to c_i are added to L_y^i to yield L_{y+1}^i. The set Γ_y^i is composed of lexicon L_y^i, acting as the set of "positive examples", plus a set of terms known not to belong to c_i, acting as the set of "negative examples".

For input to the learning device and to the term classifiers that this will eventually build, we use "bag of documents" representations for terms (Salton and McGill, 1983, pages 78–81), 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), 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. 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.

We have chosen a boosting approach not only because of its state-of-the-art effectiveness, but also because it naturally allows for a form of "data cleaning", which is useful in case a lexicographer wants to check the results and edit the newly generated lexicon. That is, in our term categorization context it allows the lexicographer to easily inspect the classified terms for possible misclassifications, since at each iteration y the algorithm, apart from generating the new lexicon L_{y+1}^i, ranks the terms in L_y^i in terms of their "hardness", i.e. how successful have been the generated classifiers at correctly recognizing their label. Since the highest ranked terms are the ones with the highest probability of having been misclassified in the previous iteration (Abney et al., 1999), the lexicographer can examine this list starting from the top and stopping where desired, removing the misclassified examples. The process of generating a thematic lexicon then becomes an iteration of generate-and-test steps.

This paper is organized as follows. In Section 2. we describe how we represent terms by means of a "bag of documents" representation.. For reasons of space we do not describe ADABOOST.MH^{KR}, the boosting algorithm we employ for term classification; see the extended paper for details (Lavelli et al., 2002). Section 3.1. discusses how to combine the indexing tools introduced in Section 2. with the boosting algorithm, and describes the role of the lexicographer in the iterative generate-and-test cycle. Section 3.2. describes the results of our preliminary experiments. In Section 4. we review related work on the automated generation of lexical resources, and spell out the differences between our and existing approaches. Section 5. concludes, pointing to avenues for improvement.

[2] We want to point out that our use of the word "term" is somehow different from the one often used in natural language processing and terminology extraction (Kageura and Umino, 1996), 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, i.e. without making any commitment as to its consisting of a single word or a sequence of words.

2. Representing terms in a space of documents

2.1. Text indexing

In text categorization applications, the process of building internal representations of texts is called *text indexing*. In text indexing, a document d_j is usually 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* and $0 \leq w_{kj} \leq 1$ represents, loosely speaking, the contribution of 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 of Tr (with α a predefined threshold, typically ranging between 1 and 5).

Different approaches to text indexing 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, which are usually removed prior to indexing) or their stems, although some researchers additionally consider noun phrases (Lewis, 1992) or "bigrams" (Caropreso et al., 2001). Different "weighting" functions may be used for tackling issue (ii), either of a probabilistic or of a statistical nature; a frequent choice is the *normalized* $tfidf$ function (see e.g. (Salton and Buckley, 1988)), which provides the inspiration for our "term indexing" methodology spelled out in Section 2.2..

2.2. Abstract indexing and term indexing

Text indexing may be viewed as a particular instance of *abstract indexing*, a task in which abstract objects are represented by means of abstract features, and whose underlying metaphor is, by and large, that the semantics of an object corresponds to the *bag of features* that "occur" in it[3]. In order to illustrate abstract indexing, let us define a *token* τ to be a specific occurrence of a given feature $f(\tau)$ in a given object $o(\tau)$, let T be the set of all tokens occurring in any of a set of objects O, and let F be the set of features of which the tokens in T are instances. Let us define the *feature frequency* $ff(f_k, o_j)$ of a feature f_k in an object o_j as

$$ff(f_k, o_j) = |\{\tau \in T \mid f(\tau) = f_k \ \wedge \ o(\tau) = o_j\}| \quad (1)$$

We next define the *inverted object frequency* $iof(f_k)$ of a feature f_k as

$$iof(f_k) = \quad (2)$$
$$= \log \frac{|O|}{|\{o_j \in O \mid \exists \tau \in T : f(\tau) = f_k \ \wedge \ o(\tau) = o_j\}|}$$

and the *weight* $w(f_k, o_j)$ of feature f_k in object o_j as

$$w_{kj} = w(f_k, o_j) = \quad (3)$$
$$= \frac{ff(f_k, o_j) \cdot iof(f_k)}{\sqrt{\sum_{s=1}^{|F|} (ff(f_s, o_j) \cdot iof(f_s))^2}}$$

We may consider the $w(f_k, o_j)$ function of Equation (3) as an *abstract indexing function*; that is, different instances of this function are obtained by specifying different choices for the set of objects O and set of features F.

The well-known text indexing function $tfidf$, mentioned in Section 2.1., is obtained by equating O with the training set of documents and F with the dictionary; T, the set of occurrences of elements of F in the elements of O, thus becomes the set of term occurrences.

Dually, a term indexing function may be obtained by switching the roles of F and O, i.e. equating F with the training set of documents and O with the dictionary; T, the set of occurrences of elements of F in the elements of O, is thus again the set of term occurrences (Schäuble and Knaus, 1992; Sheridan et al., 1997).

It is interesting to discuss the kind of intuitions that Equations (1), (2) and (3) embody in the dual cases of text indexing and term indexing:

- Equation (1) suggests that when a feature occurs multiple times in an object, the feature characterizes the object to a higher degree. In text indexing, this indicates that the more often a term occurs in a document, the more it is representative of its content. In term indexing, this indicates that the more often a term occurs in a document, the more the document is representative of the content of the term.
- Equation (2) suggests that the fewer the objects a feature occurs in, the more representative it is of the content of the objects in which it occurs. In text indexing, this means that terms that occur in too many documents are not very useful for identifying the content of documents. In term indexing, this means that the more terms a document contains (i.e. the longer it is), the less useful it is for characterizing the semantics of a term it contains.
- The intuition ("length normalization") that supports Equation (3) is that weights computed by means of $ff(f_k, o_j) \cdot iof(f_k)$ need to be normalized in order to prevent "longer objects" (i.e. ones in which many features occur) to emerge (e.g. to be scored higher in document-document similarity computations) just because of their length and not because of their content. In text indexing, this means that longer documents need to be deemphasized. In term indexing, this means instead that terms that occur in many documents need to be deemphasized[4].

It is also interesting to note that any program or data structure that implements $tfidf$ for text indexing may be used straightaway, with no modification, for term indexing: one needs only to feed the program with the terms in place of the documents and viceversa.

[3] "Bag" is used here in its set-theoretic meaning, as a synonym of *multiset*, i.e. a set in which the same element may occur several times. In text indexing, adopting a "bag of words" model means assuming that the number of times that a given word occurs in the same document is semantically significant. "Set of words" models, in which this number is assumed not significant, are thus particular instances of bag of words models.

[4] Incidentally, it is interesting to note that in switching from text indexing to term indexing, Equations (2) and (3) switch their roles: the intuition that terms occurring in many documents should be deemphasized is implemented in Equation (2) in text indexing and Equation (3) in term indexing, while the intuition that longer documents need to be deemphasized is implemented in Equation (3) in text indexing and Equation (2) in term indexing.

3. Generating thematic lexicons by bootstrapping and learning

3.1. Operational methodology

We are now ready to describe the overall process that we will follow for the generation of thematic lexicons. The process is iterative: we here describe the y-th iteration. We start from a set of thematic lexicons $L_y = \{L_y^1, \ldots, L_y^m\}$, one for each theme in $C = \{c_1, \ldots, c_m\}$, and from a corpus θ_y. We index the terms that occur in θ_y 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, the length of the vector being $r = |\theta_y|$.

By using $L_y = \{L_y^1, \ldots, L_y^m\}$ as a training set, we then generate m classifiers $\Phi_y = \{\Phi_y^1, \ldots, \Phi_y^m\}$ by applying the ADABOOST.MH^{KR} algorithm. While generating the classifiers, ADABOOST.MH^{KR} also produces, for each theme c_i, a ranking of the terms in L_y^i in terms of how hard it was for the generated classifiers to classify them correctly, which basically corresponds to their probability of being misclassified examples. The lexicographer can then, if desired, inspect L_y and remove the misclassified examples, if any (possibly rerunning, especially if these latter were a substantial number, ADABOOST.MH^{KR} on the "cleaned" version of L_y). At this point, the terms occurring in θ_y that ADABOOST.MH^{KR} has classified under c_i are added (possibly, after being checked by the lexicographer) to L_y^i, yielding L_{y+1}^i. Iteration $y+1$ can then take place, and the process is repeated again.

Note that an alternative approach is to involve the lexicographer only after the last iteration, and not after each iteration. For instance, Riloff and Shepherd (Riloff and Shepherd, 1999) perform several iterations, at each of which they add to the training set (without human intervention) the new items that have been attributed to the category with the highest confidence. After the last iteration, a lexicographer inspects the list of added terms and decides which one to remove, if any. This latter approach has the advantage of requiring the intervention of the lexicographer only once, but has the disadvantage that spurious terms added to lexicon at early iterations can cause, if not promptly removed, new spurious ones to be added in the next iterations, thereby generating a domino effect.

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, instead, 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_y is split into a training set and a test set; the classifiers are learnt 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_y. Of course, in order to guarantee a fair evaluation, the terms that never occur in θ_y are removed from the test set, since there is no way that the algorithm (or any other algorithm that extracts terms from a corpus) could possibly guess them.

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

Table 1: The contingency table for category 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.

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 taken. 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, i.e. $F_1 = \frac{2\pi\rho}{\pi+\rho}$. Effectiveness will be computed with reference to the contingency table illustrated in Table 1. When effectiveness is computed for several categories, the results for individual categories must be averaged in some way; we will do this both by *microaveraging* ("categories count proportionally to the number of their positive training examples"), i.e.

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

$$\rho^{\mu} = \frac{TP}{TP+FN} = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m}(TP_i+FN_i)}$$

and by *macroaveraging* ("all categories count the same"), i.e.

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

Here, "μ" and "M" indicate microaveraging and macroaveraging, respectively, while the other symbols are as defined in Table 1. Microaveraging rewards classifiers that behave well on *frequent categories* (i.e. categories with many positive test examples), while classifiers that perform well also on infrequent categories are emphasized by macroaveraging. Whether one or the other should be adopted obviously depends on the application.

3.3. Our experimental setting

We now describe the resources we have used in our experiments.

3.3.1. The corpora

As the corpora $\Theta = \{\theta_1, \ldots, \theta_n\}$, we have used various subsets of the Reuters Corpus Volume I (RCVI), a corpus of documents recently made available by Reuters[5] for text categorization experimentation and consisting of about 810,000 news stories. Note that, although the texts of RCVI

[5] http://www.reuters.com/

are labelled by thematic categories, we have not made use of such labels (not it would have made much sense to use them, given that these categories are different from the ones we are working with); the reasons we have chosen this corpus instead of other corpora of unlabelled texts are inessential.

3.3.2. The lexicons

As the thematic lexicons we have used subsets of an extension of WordNet, that we now describe.

WordNet (Fellbaum, 1998) is a large, widely available, non-thematic, monolingual, machine-readable dictionary in which sets of synonymous words are grouped into synonym sets (or *synsets*) organized into a directed acyclic graph. In this work, we will always refer to WordNet version 1.6.

In WordNet only a few synsets are labelled with thematic categories, mainly contained in the glosses. This limitation is overcome in WordNetDomains, an extension of WordNet described in (Magnini and Cavaglià, 2000) in which each synset has been labelled with one or more from a set of 164 thematic categories, called *domains*[6]. The 164 domains of WordNetDomains are a subset of the categories belonging to the classification scheme of Dewey Decimal Classification (DDC (Mai Chan et al., 1996)); example domains are ZOOLOGY, SPORT, and BASKETBALL.

These 164 domains have been chosen 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 field codes* (Procter, 1978)) to mark technical usages of words. Although they convey useful information for sense discrimination, they typically tag only a small portion of a dictionary. WordNetDomains extends instead the coverage of domain labels to an entire, existing lexical database, i.e. WordNet.

A domain may include synsets of different syntactic categories: 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 senses of the same word into thematic clusters, with the side effect of reducing word polysemy in WordNet.

The annotation methodology used in (Magnini and Cavaglià, 2000) for creating WordNetDomains was mainly manual, and based on lexico-semantic criteria which take advantage from the already existing conceptual relations in WordNet. First, a small number of high level synsets were manually annotated with their correct domains. Then, an automatic procedure exploiting some of the WordNet relations (i.e. hyponymy, troponymy, meronymy, antonymy and pertain-to) was used in order to extend these assignments to all the synsets reachable through inheritance. For example, this procedure automatically marked the synset {`beak, bill, neb, nib`} with the code ZOOLOGY, starting from the fact that the synset {`bird`} was itself tagged with ZOOLOGY, and following a "part-of" relation (one of the meronymic relations present in WordNet). In some cases the inheritance procedure had to be manually blocked, inserting an "exception" in order to prevent a wrong propagation. For instance, if blocking had not been used, the term `barber_chair#1`, being a "part-of" `barbershop#1`, which is annotated with COMMERCE, would have inherited COMMERCE, which is unsuitable.

For the purpose of the experiments reported in this paper, we have used a simplified variant of WordNetDomains, called WordNetDomains(42). This was obtained from WordNetDomains by considering only 42 highly relevant labels, 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 (that is, 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 both the synsets that it originally labelled in WordNetDomains, plus the ones that in WordNetDomains were labelled under its children categories (e.g. VOLLEY, BASKETBALL, ...) or under its parent category (FREE-TIME). Since FREE-TIME has another child (PLAY) which is also retained in WordNetDomains(42), the synsets originally labelled by FREE-TIME will now be labelled also by PLAY, and will thus have multiple labels. However, that a synset may have multiple labels is true in general, i.e. these labels need not have any particular relation in the hierarchy.

This restriction to the 42 most significant categories allows to obtain a good compromise between the conflicting needs of avoiding data sparseness and preventing the loss of relevant semantic information. These 42 categories belong to 5 groups, where the categories in a given group are all the children of the same WordNetDomains category, 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 into WordNetDomains(42)).

3.3.3. The experiment

We have run several experiments for different choices of the subset of RCVI chosen as corpus of text θ_y, and for different choices of the subsets of WordNetDomains(42) chosen as training set Tr_y and test set Te_y. We first describe how we have run a generic experiment, and then go on to describe the sequence of different experiments we have run. For the moment being we have run experiments consisting of one iteration only of the bootstrapping process. In future experiments we also plan to allow for multiple iterations, in which the system learns new terms also from previously learnt ones.

In our experiments we considered only nouns, thereby discarding words tagged by other syntactic categories. We plan to also consider words other than nouns in future ex-

[6] 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 *multi-label* categorization task (Sebastiani, 2002, Section 2.2).

periments.

For each experiment, we discarded all documents that did not contain any term from the training lexicon Tr_y, since they do not contribute in representing the meaning of training documents, and thus could not possibly be of any help in building the classifiers. Next, we discarded all "empty" training terms, i.e. training terms that were not contained in any document of θ_y, since they could not possibly contribute to learning the classifiers. Also empty test terms were discarded, since no algorithm that extracts terms from corpora could possibly extract them. Quite obviously, we also do not use the terms that occur in θ_y but belong neither to the training set Tr_y nor to the test set Te_y.

We then lemmatized all remaining documents and annotated the lemmas with part-of-speech tags, both by means of the TREETAGGER package (Schmid, 1994); we also used the WordNet morphological analyzer in order to resolve ambiguities and lemmatization mistakes. After tagging, we applied a filter in order to identify the words actually contained in WordNet, including multiwords, and then we discarded all terms but nouns. The final set of terms that resulted from this process was randomly divided into a training set Tr_y (consisting of two thirds of the entire set) and a test set Te_y (one third). As negative training examples of category c_i we chose all the training terms that are not positive examples of c_i.

Note that in this entire process we have not considered the grouping of terms into synsets; that is, the lexical units of interest in our application are the terms, and not the synsets. The reason is that RCVI is not a sense-tagged corpus, and for any term occurrence τ it is not clear to which synset τ refers to.

3.3.4. The results

Our experimental results on this task are still very preliminary, and are reported in Table 2.

Instead of tackling the entire RCVI corpus head on, for the moment being we have run only small experiments on limited subsets of it (up to 8% of its total size), with the purpose of getting a feel for which are the dimensions of the problem that need investigation; for the same reason, for the moment being we have used only a small number of boosting iterations (500). In Table 2, the first three lines concern experiments on the news stories produced on a single day (08.11.1996); the next three lines use the news stories produced in a single week (08.11.1996 to 14.11.1996), and the last six lines use the news stories produced in an entire month (01.11.1996 to 30.11.1996). Only training and test terms occurring in at least x documents were considered; the experiments reported in the same block of lines differ for the choice of the x parameter.

There are two main conclusions we can draw from these still preliminary experiments. The first conclusion is that 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 (Ault and Yang, 2001); a lot of work is still needed in tuning this approach in order to obtain significant categorization performance. The low values of F_1 are mostly the result of low recall values, while precision tends to be much higher, often well above the 70% mark. Note that the low absolute performance might also be explained, at least partially, with the imperfect quality of the WordNetDomains(42) resource, which was generated by a combination of automatic and manual procedures and did no undergo extensive checking afterwards.

The second conclusion is that results show a constant and definite improvement when higher values of x are used, despite the fact that higher levels of x mean a higher number of labels per term, i.e. more polysemy. This is not surprising, since when a term occurs e.g. in one document only, this means that only one entry in the vector that represents the term is non-null (i.e. significant). This is in sharp contrast with text categorization, in which the number of non-null 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. This alone might suffice to justify the difference in performance between term categorization and text categorization.

However, one reason the actual F_1 scores are low is that this is a hard task, and the evaluation standards we have adopted are considerably tough. This is discussed in the next paragraph.

No baseline? Note that we present no baseline, either published or new, against which to compare our results, for the simple fact that term categorization as we conceive it here is a novel task, and there are as yet no previous results or known approaches to the problem to compare with.

Only (Riloff and Shepherd, 1999; Roark and Charniak, 1998) have approached the problem of extending an existing thematic lexicon with new terms drawn from a text corpus. However, there are key differences between their evaluation methodology and ours, which makes comparisons difficult and unreliable. First, their "training" terms have not been chosen randomly our of a thematic 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; Roark and Charniak, 1998) extract the terms from texts that are known to be about the theme, which makes the task easier than ours; conversely, by using generic texts, we avoid the costly process of labelling the documents by thematic categories, and we are able to generate thematic lexicons for multiple themes at once *from the same unlabelled text corpus*. Third, their evaluation methodology is manual, i.e. subjective, in the sense that the authors themselves manually checked the results of their experiments, judging, for each returned term, how reasonable the inclusion of the term in the lexicon is[7]. This sharply contrasts with our evaluation methodology, which is completely automatic (since we measure the proficiency

[7] For instance, (Riloff and Shepherd, 1999) judged a word classified into a category correct also if they judged that "the word refers to a part of a member of the category", thereby judging the words `cartridge` and `clips` to belong to the domain WEAPONS. This looks to us a loose notion of category mambership, and anyway points to the pitfalls of "subjective" evaluation methodologies.

# of docs	# of training terms	# of test terms	# of labels per term	minimum # of docs per term	Precision micro	Recall micro	F_1 micro	Precision macro	Recall macro	F_1 macro
2,689	4,424	2,212	1.96	1	0.542029	0.043408	0.080378	0.584540	0.038108	0.071551
2,689	1,685	842	2.36	5	0.512903	0.079580	0.137782	0.487520	0.078677	0.135489
2,689	1,060	530	2.55	10	0.517544	0.086131	0.147685	0.560876	0.084176	0.146383
16,003	7,975	3,987	1.76	1	0.720165	0.049631	0.092863	0.701141	0.038971	0.073837
16,003	4,132	2,066	2.02	5	0.733491	0.075121	0.136284	0.738505	0.065472	0.120281
16,003	2,970	1,485	2.15	10	0.740260	0.091405	0.162718	0.758044	0.078162	0.141712
67,953	11,313	5,477	1.66	1	0.704251	0.043090	0.081211	0.692819	0.034241	0.065256
67,953	6,829	3,414	1.83	5	0.666667	0.040816	0.076923	0.728300	0.050903	0.095155
67,953	5,335	2,668	1.92	10	0.712406	0.076830	0.138701	0.706678	0.056913	0.105342
67,953	4,521	2,261	1.99	15	0.742574	0.086445	0.154863	0.731530	0.064038	0.117766
67,953	3,317	1,659	2.10	30	0.745455	0.098439	0.173913	0.785371	0.075573	0.137878
67,953	2,330	1,166	2.25	60	0.760417	0.117789	0.203982	0.755136	0.086809	0.155718

Table 2: Preliminary results obtained on the automated lexicon generation task (see Section 3.3. for details).

of our system at discovering terms about the theme, by the capability of the system to replicate the lexicon generation work of a lexicographer), can be replicated by other researchers, and is unaffected by possible experimenter's bias. Fourth, checking one's results for "reasonableness", as (Riloff and Shepherd, 1999; Roark and Charniak, 1998) do, means that one can only ("subjectively") measure precision (i.e. whether the terms spotted by the algorithm do in fact belong to the theme), but not recall (i.e. whether the terms belonging to the theme have actually been spotted by the algorithm). Again, this is in sharp contrast with our methodology, which ("objectively") measures precision, recall, and a combination of them. Also, note that in terms of precision, i.e. the measure that (Riloff and Shepherd, 1999; Roark and Charniak, 1998) subjectively compute, our algorithm fares pretty well, mostly scoring higher than 70% even in these very preliminary experiments.

4. Related work

4.1. Automated generation of lexical resources

The automated generation of lexicons from text corpora has a long history, dating back at the very least to the seminal works of Lesk, Salton and Sparck Jones (Lesk, 1969; Salton, 1971; Sparck Jones, 1971), and has been the subject of active research throughout the last 30 years, both within the information retrieval 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 *cluster-based thesauri*, i.e. networks of groups of synonymous or quasi-synonymous words, in which 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 values. When the lexical resources being built are of a *thematic* nature, the thematic nature of a word is usually established by checking whether its frequency within thematic documents is higher than its frequency in generic documents (Chen et al., 1996; Riloff and Shepherd, 1999; Schatz et al., 1996; Sebastiani, 1999) (this property is often called *salience* (Yarowsky, 1992)).

In the approach described above, the key decision is how to tackle step (i), and there are two main approaches to this. In the first approach the similarity between two words is usually computed in terms of their degree of co-occurrence and co-absence within the same document (Crouch, 1990; Crouch and Yang, 1992; Qiu and Frei, 1993; Schäuble and Knaus, 1992; Sheridan and Ballerini, 1996; Sheridan et al., 1997); variants of this approach are obtained by restricting the context of co-occurrence from the document to the paragraph, or to the sentence (Schütze, 1992; Schütze and Pedersen, 1997), or to smaller linguistic units (Riloff and Shepherd, 1999; Roark and Charniak, 1998). In the second approach this similarity is computed from head-modifier structures, by relying on the assumption that frequent modifiers of the same word are semantically similar (Grefenstette, 1992; Ruge, 1992; Strzalkowski, 1995). The latter approach can also deal with indirect co-occurrence[8], but the former is conceptually simpler, since it does not even need any parsing step.

This literature (apart from (Riloff and Shepherd, 1999; Roark and Charniak, 1998), which are discussed below) has thus taken an *unsupervised* learning approach, which can be summarized in the recipe "from a set of documents about theme t and a set of generic documents (i.e. mostly not about t), extract the words that mostly characterize t". Our work is different, in that its underlying *supervised* learning approach requires a starting kernel of terms about t, but does not require that the corpus of documents from which

[8]We say that words w_1 and w_2 *co-occur directly* when they both occur in the same document (or other linguistic context), while we say that they *co-occur indirectly* when, for some other word w_3, w_1 and w_3 co-occur directly and w_2 and w_3 co-occur directly. Perfect synonymy is not revealed by direct co-occurrence, since users tend to consistently use either one or the other synonym but not both, while it is obviously revealed by indirect co-occurrence. However, this latter also tends to reveal many more "spurious" associations than direct co-occurrence.

the terms are extracted be labelled. This makes our supervised technique particularly suitable for *extending* a previously existing thematic 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 thematic lexical resource by some unsupervised technique, and then (ii) extending it by our supervised technique. An intermediate approach between these two is the one adopted in (Riloff and Shepherd, 1999; Roark and Charniak, 1998), which also requires a starting kernel of terms about t, but also requires a set of documents about theme t from which the new terms are extracted.

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

A work which is closer in spirit to ours than the abovementioned ones is (Tokunaga et al., 1997), since it deals with using previously classified terms as training examples in order to classify new terms. This work exploits a naive Bayesian model for classification in conjunction with another learning method, chosen among nearest neighbour, "category-based" (by which the authors basically mean a Rocchio method – see e.g. (Sebastiani, 2002, Section 6.7)) and "cluster-based" (which does not use category labels of training examples). However, these latter learning methods and (especially) the nature of their integration with the naive Bayesian model are not specified in mathematical detail, which does not allow us to make a precise comparison between the model of (Tokunaga et al., 1997) and ours. Anyway, our model is more elegant, in that it just assumes a single learning method (for which we have chosen boosting, although we might have chosen any other supervised learning method), and in that it replaces the ad-hoc notion of "co-occurrence" with a theoretically sounder "dual" theory of text indexing, which allows one, among other things, to bring to bear any kind of intuitions on term weighting, or any kind of text indexing theory, that are known from information retrieval.

4.2. Boosting

Boosting has been applied to several learning tasks related to text analysis, including POS-tagging and PP-attachment (Abney et al., 1999), clause splitting (Carreras and Màrquez, 2001b), word segmentation (Shinnou, 2001), word sense disambiguation (Escudero et al., 2000), text categorization (Schapire and Singer, 2000; Schapire et al., 1998; Sebastiani et al., 2000; Taira and Haruno, 2001), e-mail filtering (Carreras and Márquez, 2001a), document routing (Iyer et al., 2000; Kim et al., 2000), and term extraction (Vivaldi et al., 2001). Among these works, the one somehow closest in spirit to ours is (Vivaldi et al., 2001), since it is concerned with extracting medical terms from a corpus of texts. A key difference with our work is that the features by which candidate terms are represented in (Vivaldi et al., 2001) are not simply the documents they occur in, but the results of term extraction algorithms; therefore, our approach is simpler and more general, since it does not require the existence of separate term extraction algorithms.

5. Conclusion

We have reported work in progress on the semi-automatic generation of thematic lexical resources by the combination of (i) a dual interpretation of IR-style text indexing theory and (ii) a boosting-based machine learning approach. Our method does not require pre-existing semantic knowledge, and is particularly suited to the situation in which one or more preexisting thematic lexicons need to be extended and no corpora of texts classified according to the themes are available. We have run only initial experiments, which suggest that the approach is viable, although large margins of improvement exist. In order to improve the overall performance we are planning several modifications to our currently adopted strategy.

The first modification consists in performing *feature selection*, as commonly used in text categorization (Sebastiani, 2002, Section 5.4). This will consist in individually scoring (by means of the *information gain* function) all documents in terms of how indicative they are of the occurrence or non-occurrence of the categories we are interested in, and to choose only the best-scoring ones out of a potentially huge corpus of available documents.

The second avenue we intend to follow consists in trying alternative notions of what a document is, by considering as "documents" paragraphs, or sentences, or even smaller, syntactically characterized units (as in (Riloff and Shepherd, 1999; Roark and Charniak, 1998)), rather than full-blown Reuters news stories.

A third modification consists in selecting, as the negative examples of a category c_i, all the training examples that are not positive examples of c_i *and* are at the same time positive examples of (at least one of) the siblings of c_i. This method, known as the *query-zoning method* or as the *method of quasi-positive examples*, is known to yield superior performance with respect to the method we currently use (Dumais and Chen, 2000; Ng et al., 1997).

The last avenue for improvement is the optimization of the parameters of the boosting process. The obvious parameter that needs to be optimized is the number of boosting iterations, which we have kept to a minimum in the reported experiments. A less obvious parameter is the form of the initial distribution on the training examples (that we have not described here for space limitations); by changing it with respect to the default value (the uniform distribution) we will be able to achieve a better compromise between precision and recall (Schapire et al., 1998), which for the moment being have widely different values.

Acknowledgments

We thank Henri Avancini for help with the coding task and Pio Nardiello for assistance with the

ADABOOST.MH^{KR} code. Above all, we thank Roberto Zanoli for help with the coding task and for running the experiments.

6. References

Steven Abney, Robert E. Schapire, and Yoram Singer. 1999. Boosting applied to tagging and PP attachment. In *Proceedings of EMNLP-99, 4th Conference on Empirical Methods in Natural Language Processing*, pages 38–45, College Park, MD.

Thomas Ault and Yiming Yang. 2001. kNN, Rocchio and metrics for information filtering at TREC-10. In *Proceedings of TREC-10, 10th Text Retrieval Conference*, Gaithersburg, US.

Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. 2001. 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.

Xavier Carreras and Lluís Márquez. 2001a. Boosting trees for anti-spam email filtering. In *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG.

Xavier Carreras and Lluís Màrquez. 2001b. Boosting trees for clause splitting. In *Proceedings of CONLL-01, 5th Conference on Computational Natural Language Learning*, Toulouse, FR.

Hsinchun Chen, Chris Schuffels, and Rich Orwing. 1996. Internet categorization and search: A machine learning approach. *Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries*, 7(1):88–102.

Carolyn J. Crouch and Bokyung Yang. 1992. Experiments in automated statistical thesaurus construction. In *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 77–87, Kobenhavn, DK.

Carolyn J. Crouch. 1990. An approach to the automatic construction of global thesauri. *Information Processing and Management*, 26(5):629–640.

Susan T. Dumais and Hao Chen. 2000. Hierarchical classification of Web content. In *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, GR. ACM Press, New York, US.

Gerard Escudero, Lluís Màrquez, and German Rigau. 2000. Boosting applied to word sense disambiguation. In *Proceedings of ECML-00, 11th European Conference on Machine Learning*, pages 129–141, Barcelona, ES.

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

Gregory Grefenstette. 1992. Use of syntactic context to produce term association lists for retrieval. In *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 89–98, Kobenhavn, DK.

Gregory Grefenstette. 1994. *Explorations in automatic thesaurus discovery*. Kluwer Academic Publishers, Dordrecht, NL.

Lynette Hirschman, Ralph Grishman, and Naomi Sager. 1988. Grammatically-based automatic word class formation. *Information Processing and Management*, 11(1/2):39–57.

Raj D. Iyer, David D. Lewis, Robert E. Schapire, Yoram Singer, and Amit Singhal. 2000. Boosting for document routing. In *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 70–77, McLean, US.

Yufeng Jing and W. Bruce Croft. 1994. An association thesaurus for information retrieval. In *Proceedings of RIAO-94, 4th International Conference "Recherche d'Information Assistee par Ordinateur"*, pages 146–160, New York, US.

Kyo Kageura and Bin Umino. 1996. Methods of automatic term recognition: a review. *Terminology*, 3(2):259–289.

Yu-Hwan Kim, Shang-Yoon Hahn, and Byoung-Tak Zhang. 2000. Text filtering by boosting naive Bayes classifiers. In *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 168–75, Athens, GR.

Alberto Lavelli, Bernardo Magnini, and Fabrizio Sebastiani. 2002. Building thematic lexical resources by term categorization. Technical report, Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT. Forthcoming.

Michael E. Lesk. 1969. Word-word association in document retrieval systems. *American Documentation*, 20(1):27–38.

David D. Lewis. 1992. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK.

Bernardo Magnini and Gabriela Cavaglià. 2000. Integrating subject field codes into WordNet. In *Proceedings of LREC-2000, 2nd International Conference on Language Resources and Evaluation*, pages 1413–1418, Athens, GR.

Lois Mai Chan, John P. Comaromi, Joan S. Mitchell, and Mohinder Satija. 1996. *Dewey Decimal Classification: a practical guide*. OCLC Forest Press, Albany, US, 2nd edition.

Hwee T. Ng, Wei B. Goh, and Kok L. Low. 1997. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 67–73, Philadelphia, US. ACM Press, New York, US.

Helen J. Peat and Peter Willett. 1991. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5):378–383.

Paul Procter, editor. 1978. *The Longman Dictionary of Contemporary English*. Longman, Harlow, UK.

Yonggang Qiu and Hans-Peter Frei. 1993. Concept-based query expansion. In *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Devel-*

opment in Information Retrieval, pages 160–169, Pittsburgh, US.

Ellen Riloff and Jessica Shepherd. 1999. A corpus-based bootstrapping algorithm for semi-automated semantic lexicon construction. *Journal of Natural Language Engineering*, 5(2):147–156.

Brian Roark and Eugene Charniak. 1998. Noun phrase cooccurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of ACL-98, 36th Annual Meeting of the Association for Computational Linguistics*, pages 1110–1116, Montreal, CA.

Gerda Ruge. 1992. Experiments on linguistically-based terms associations. *Information Processing and Management*, 28(3):317–332.

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

Gerard Salton and Michael J. McGill. 1983. *Introduction to modern information retrieval*. McGraw Hill, New York, US.

Gerard Salton. 1971. Experiments in automatic thesaurus construction for information retrieval. In *Proceedings of the IFIP Congress*, volume TA-2, pages 43–49, Ljubljana, YU.

Robert E. Schapire and Yoram Singer. 2000. BOOSTEXTER: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Robert E. Schapire, Yoram Singer, and Amit Singhal. 1998. Boosting and Rocchio applied to text filtering. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 215–223, Melbourne, AU.

Bruce R. Schatz, Eric H. Johnson, Pauline A. Cochrane, and Hsinchun Chen. 1996. Interactive term suggestion for users of digital libraries: Using subject thesauri and co-occurrence lists for information retrieval. In *Proceedings of DL-96, 1st ACM Digital Library Conference*, pages 126–133, Bethesda, US.

Peter Schäuble and Daniel Knaus. 1992. The various roles of information structures. In *Proceedings of the 16th Annual Conference of the Gesellschaft für Klassifikation*, pages 282–290, Dortmund, DE.

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

Hinrich Schütze and Jan O. Pedersen. 1997. A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing and Management*, 33(3):307–318.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings of Supercomputing'92*, pages 787–796, Minneapolis, US.

Fabrizio Sebastiani, Alessandro Sperduti, and Nicola Valdambrini. 2000. An improved boosting algorithm and its application to automated text categorization. In *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 78–85, McLean, US.

Fabrizio Sebastiani. 1999. Automated generation of category-specific thesauri for interactive query expansion. In *Proceedings of IDC-99, 9th International Database Conference on Heterogeneous and Internet Databases*, pages 429–432, Hong Kong, CN.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Páraic Sheridan and Jean-Paul Ballerini. 1996. Experiments in multilingual information retrieval using the SPIDER system. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 58–65, Zürich, CH.

Páraic Sheridan, Martin Braschler, and Peter Schäuble. 1997. Cross-language information retrieval in a multilingual legal domain. In *Proceedings of ECDL-97, 1st European Conference on Research and Advanced Technology for Digital Libraries*, pages 253–268, Pisa, IT.

Hiroyuki Shinnou. 2001. Detection of errors in training data by using a decision list and AdaBoost. In *Proceedings of the IJCAI-01 Workshop on Text Learning: Beyond Supervision*, Seattle, US.

Karen Sparck Jones. 1971. *Automatic keyword classification for information retrieval*. Butterworths, London, UK.

Tomek Strzalkowski. 1995. Natural language information retrieval. *Information Processing and Management*, 31(3):397–417.

Hirotoshi Taira and Masahiko Haruno. 2001. Text categorization using transductive boosting. In *Proceedings of ECML-01, 12th European Conference on Machine Learning*, pages 454–465, Freiburg, DE.

Takenobu Tokunaga, Makoto Iwayama, and Hozumi Tanaka. 1995. Automatic thesaurus construction based on grammatical relations. In *Proceedings of IJCAI-95, 14th International Joint Conference on Artificial Intelligence*, pages 1308–1313, Montreal, CA.

Takenobu Tokunaga, Atsushi Fujii, Makoto Iwayama, Naoyuki Sakurai, and Hozumi Tanaka. 1997. Extending a thesaurus by classifying words. In *Proceedings of the ACL-EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources*, pages 16–21, Madrid, ES.

Jordi Vivaldi, Lluís Màrquez, and Horacio Rodríguez. 2001. Improving term extraction by system combination using boosting. In *Proceedings of ECML-01, 12th European Conference on Machine Learning*, pages 515–526, Freiburg, DE.

David Yarowsky. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of COLING-92, 14th International Conference on Computational Linguistics*, pages 454–460, Nantes, FR.