Distributional Term Representations: An Experimental Comparison*

Alberto Lavelli
ITC-irst
Via Sommarive 18
38050 Povo di Trento, Italy
lavelli@itc.it

Fabrizio Sebastiani†
ISTI-CNR
Via Giuseppe Moruzzi 1
56124 Pisa, Italy
fabrizio.sebastiani@isti.cnr.it

Roberto Zanoli
ITC-irst
Via Sommarive 18
38050 Povo di Trento, Italy
zanoli@itc.it

ABSTRACT

A number of content management tasks, including term categorization, term clustering, and automated thesaurus generation, view natural language *terms* (e.g. words, noun phrases) as first-class objects, i.e. as objects endowed with an internal representation which makes them suitable for explicit manipulation by the corresponding algorithms. The information retrieval (IR) literature has traditionally used an extensional (aka *distributional*) representation for terms according to which a term is represented by the "bag of documents" in which the term occurs. The computational linguistics (CL) literature has independently developed an alternative distributional representation for terms, according to which a term is represented by the "bag of terms" that co-occur with it in some document. This paper aims at discovering which of the two representations is most effective, i.e. brings about higher effectiveness once used in tasks that require terms to be explicitly represented and manipulated. We carry out experiments on (i) a term categorization task, and (ii) a term clustering task; this allows us to compare the two different representations in closely controlled experimental conditions. We report the results of experiments in which we categorize/cluster under 42 different classes the terms extracted from a corpus of more than 65,000 documents. Our results show a substantial difference in effectiveness between the two representation styles; we give both an intuitive explanation and an information-theoretic justification for these different behaviours.

1. INTRODUCTION

Many traditional information retrieval (IR) tasks, such as text search, text clustering, or text categorization (aka "classification"), have natural language documents as their first-class objects, in the sense that the algorithms that are meant to solve these tasks require explicit internal representations of the documents they need to deal with. In IR documents are usually given an extensional (aka *distributional*), vectorial representation, in which the dimensions (aka *features*) of the vector representing a document are the *terms* occurring in the document. Here, "term" is a neutral expression denoting whatever entity is deemed to constitute the atomic unit of meaning in a text; depending on the choice of the designer, terms may be words, "stems" (i.e. the morphological roots of words), lemmata, noun phrases, n-grams, or other.

In this paper we deal with content management tasks whose first-class objects are terms, and not documents. Examples of such tasks (that we dub *term management tasks*) are term categorization [1] (i.e. grouping terms, according to their meaning, into prespecified classes), term clustering [20] (i.e. grouping terms, according to their meaning, into groups not known in advance), automated thesaurus generation [4] (i.e. extracting a thesaurus from a corpus of either generic of domain-specific texts), or word sense disambiguation [12] (i.e. choosing the semantic sense of a word occurrence from a predefined list of senses of this word). All these tasks, which fall at the crossroads of IR and computational linguistics (CL), are based on an "understanding" of the meaning of a term, and thus require a representation of this meaning that be amenable to interpretation by the algorithms that carry out these tasks.

The approach to term representation that the IR community has almost universally adopted is a natural variant of the above-mentioned approach for document representation, that the very same community developed. In this approach, the features of the vectors that represent terms are *documents*. The underlying metaphor is that, as the semantics of a document is conveyed by the terms that occur in it, the semantics of a term is conveyed by the documents in which the term occurs.

The CL community, instead, has developed a largely independent stream of research based on a different, albeit related, distributional representation. Here, the features of the vectors that represent terms are other *terms*, the underlying metaphor being that the semantics of a term is conveyed by the terms that co-occur with it (i.e. that occur

*This is a revised and much extended version of a paper titled "An Experimental Comparison of Term Representations for Term Management Applications" forthcoming at SEBD-04, the 2004 Italian Workshop on Advanced Database Systems.

†To whom all correspondence should be addressed.

CIKM '04, Washington, US

in the same documents). The basic intuition behind this representation (and, as we will discuss in Section 3, behind the previously discussed representation too) is the so-called *distributional hypothesis*, formulated by the well-known linguist Zellig Harris [16], which states that terms with similar distributional patterns tend to have the same meaning[1].

The relative merits of these two alternative representations have never, to the best of our knowledge, been assessed. The aim of this paper is thus to compare the two representations experimentally. Since there are no metrics for evaluating the goodness of a vectorial representation, the evaluation can only be "extrinsic" [14], i.e. obtained by testing a given system, fed with terms represented according to one or the other method, on one or more tasks for which evaluation metrics are defined.

In this paper we perform an extrinsic evaluation of these two representations by applying them to the tasks of term categorization and clustering. The reason we have chosen these two tasks is that there are standard, reliable, well-accepted evaluation metrics for them, while the same cannot be said of other tasks such as automated thesaurus construction.

Term categorization (see [1] for more details) is a supervised machine learning problem, since it involves learning a term classifier from a training set Tr of terms preclassified under a set of categories $C = \{c_1, \ldots, c_m\}$, where the terms are implicitly represented by their occurrence in the documents of a corpus Δ. Effectiveness testing is achieved by applying the classifier to a test set Te of terms preclassified under C and measuring the degree of coincidence between the classes attributed by the classifier and those originally attached to the test terms.

Term clustering consists instead of grouping a set of terms into m groups so that terms with similar meaning fall in the same group. Effectiveness testing may be achieved by clustering into m groups a set of terms pre-classified under a set of m categories $C = \{c_1, \ldots, c_m\}$, and measuring the degree of coincidence between C and the generated cluster structure.

The structure of this paper is as follows. In Section 2 we illustrate the two alternative representations for terms that this paper aims to compare. Section 3 presents the term categorization and term clustering experiments we have run in order to "extrinsically" evaluate the relative level of effectiveness of the two representations. Section 4 discusses the results of these experiments, and presents (i) an intuitive argument and (ii) an information-theoretic argument that explain them. Section 5 discusses related work, while Section 6 concludes.

2. ALTERNATIVE VECTORIAL REPRESENTATIONS FOR TERMS

2.1 Representing documents

The IR community has a long-standing tradition in term management applications, such as term clustering [19, 20, 24, 42, 43] or automated thesaurus construction [4, 6, 7, 28, 31, 33, 40, 41], that make use of explicit representations for terms. 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, known as *the bag-of-words approach*, 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 of 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 paper is[2]

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

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

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

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

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

2.2 The document occurrence representation

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*

[1] The famous linguist J.R. Firth [11] expressed a similar vision in his often-quoted motto "You shall know a word by the company it keeps".

[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 paper are independent of the weighting function used, and any other function could have been used for this purpose.

$\vec{t_j} = \langle w_{1j}, \ldots, w_{rj} \rangle$, 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[3]. 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 Section 2.1, now aptly renamed the $dfitf$ function, is reinterpreted as follows:

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

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 too normalized by cosine normalization, finally yielding

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

Symmetrically, here 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.

2.3 The term co-occurrence representation

The CL community too has a long-standing tradition in term management applications such as term clustering [3, 10, 27], word sense disambiguation [12, 13, 35, 36], or automated thesaurus construction [15, 37]. However, this tradition has developed largely independently of the IR term management tradition, and has given rise to a different style of term representation, that we will here call the *term co-occurrence representation* (TCOR) [8]. This representation is, like the DOR, of a distributional and vectorial nature; however, the basic idea that underlies it is that the semantics of a term t_j may be viewed as coinciding with the bag of terms that co-occur with t_j in the documents belonging to the document collection $\mathcal{D}$. Here, a term t_j is represented by a vector $\vec{t_j} = \langle w_{1j}, \ldots, w_{rj} \rangle$ of weighted *terms*, where r is the cardinality of the dictionary $\mathcal{T}$ (defined as in Section 2.1) and the weight $0 \leq w_{kj} \leq 1$ represents the contribution of t_k to the specification of the semantics of t_j.

Note that, while in the DOR the representation of the $|\mathcal{T}|$ terms that occur in the $|\mathcal{D}|$ documents constituting the collection consisted in a $|\mathcal{T}| \times |\mathcal{D}|$ matrix, in the TCOR we have a $|\mathcal{T}| \times |\mathcal{T}|$ square matrix in which the elements on the diagonal all have a value of 1 (since any term always co-occurs with itself).

The $tfidf$ function of Section 2.1 can be reinterpreted in terms of the TCOR too. Now aptly renamed $tfitf$, its form is:

$$tfitf(t_k, t_j) = tf(t_k, t_j) \cdot \log \frac{|\mathcal{T}|}{\#_{\mathcal{T}}(t_k)} \quad (5)$$

where $\#_{\mathcal{T}}(t_k)$ denotes the number of terms in the dictionary $\mathcal{T}$ which co-occur with t_k in at least one document and

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

where $\#(t_k, t_j)$ denotes the number of documents in which t_k and t_j co-occur. As usual, weights obtained by Equation (5) are normalized by cosine normalization, finally yielding

$$w_{kj} = \frac{tfitf(t_k, t_j)}{\sqrt{\sum_{s=1}^{|\mathcal{T}|} tfitf(t_s, t_j)^2}} \quad (6)$$

The intuitions underlying this weighting function are that (a) the more documents t_k and t_j co-occur in, the more important t_k is for characterizing the semantics of t_j; (b) the more distinct terms t_k co-occurs with, the smaller its contribution is in characterizing the semantics of a term t_i with which it co-occurs.

We should also add that many variants of this policy have been proposed which differ in terms of how the notion of "document" (i.e. the linguistic context, or text window, in which the co-occurrence between two terms t_1 and t_2 is considered significant, which we will here call the *reference context*) is interpreted. While some approaches indeed use the entire document as the reference context, others restrict it to a sliding, fixed-size window of text centered around the focus term [22, 30]. Other authors instead 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, [21, 26] represents 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.

3. EXPERIMENTS

The DOR and the TCOR seem equally plausible representations for terms, and based on equally plausible intuitions on what "distributionally" contributes to determining the semantics of a term. Note that term co-occurrence is the fundamental notion that underlies *both* representations. While this is explicit in the TCOR, it is nonetheless the case in the DOR too. To see this, note that in the DOR two terms have the same representation when they occur exactly in the same documents and exactly the same number of times in each of the documents in which they occur, and they have similar representations when they simply *tend* to do so. Still, the DOR and the TCOR are different, and (as we will see in this section) they bring about different levels of effectiveness once used in practical applications.

In order to see this, we perform an extrinsic evaluation of these two representations by applying them to the tasks of term categorization and term clustering, respectively. The rest of this section is devoted to describing the experiments we have run on these two tasks.

3.1 Term categorization experiments

[3]It may be worthwhile to mention that the term-document matrix resulting from pulling together all the (binary, i.e. only using presence-absence of the term in the document) vectors that represent terms is at the heart of the *latent semantic analysis* approach to document indexing [9]. This may be seen as a more sophisticated version of term clustering in which, instead of k clusters of terms, k different linear combinations of all the terms are produced.

Term categorization is the task of classifying terms according to a predefined set of thematic categories $C = \{c_1, \ldots, c_m\}$. This is accomplished by learning a term classifier from a training set Tr of terms preclassified under C, where the terms are implicitly represented by their occurrence in the documents of a corpus Δ. Effectiveness testing is achieved by applying the classifier to a test set Te of terms preclassified under C and measuring the degree of coincidence between the classes attributed by the classifier and those originally attached to the test terms.

3.1.1 The experimental setting

Two datasets are required for testing the effectiveness of a given term categorization method: a set Θ of terms preclassified according to the set $C = \{c_1, \ldots, c_m\}$ of categories, and a set Δ of documents from which the representations of the terms must be extracted.

As for the set Θ, we have chosen WordNetDomains(42) [1], a lexical resource in which each term in WordNet (version 1.6) is labelled according to a set of 42 very general semantic categories (examples of which are ADMINISTRATION, AGRICULTURE, ALIMENTATION, ...). WordNetDomains(42) is actually just a coarser-grained version of WordNetDomains [23]; in the latter the categories are 164 and are finer-grained (see [1] for details on the relationship between WordNetDomains(42) and WordNetDomains). The terms we have considered in our experiments are just the nouns, i.e. we have discarded the 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 categories, since they tend to be more category-specific than e.g. verbs or adverbs.

As the corpus Δ we have randomly chosen a subset of Reuters Corpus Volume 1 (RCV1)[4], consisting of the 67,953 news stories produced by Reuters from 1 Nov 1996 to 30 Nov 1996 (the original RCV1 contains all the 806,812 news stories from 20 Aug 1996 to 19 Aug 1997). All RCV1 news stories are in English, and have roughly 110 distinct terms per document on average [29].

Before running the experiments we have lemmatized all the documents in the corpus Δ and annotated the lemmas with part-of-speech tags, both by means of the TREETAGGER package [34]. We have also used the WordNet morphological analyzer in order to resolve ambiguities and lemmatization mistakes. During this phase we have also performed the recognition of the 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 have performed a *term filtering* phase, in which we have discarded:

- all "empty" terms, i.e. WordNetDomains(42) terms that are not contained in any document of the corpus Δ, 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 Δ but do not belong to WordNetDomains(42), since they do not play any role in our experiments.

[4] `http://www.reuters.com/`

After this term filtering phase there are 16,790 nouns left in WordNetDomains(42). We have repeated each term categorization 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 Figure 1 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[5].

We have randomly divided the set of the 16,790 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 have chosen all the training terms that are not positive examples of c_i. Finally, before learning the term classifiers, we have performed a *document filtering* phase by discarding all documents that do not contain any term from Tr, since they do not contribute to represent the meaning of training terms, and thus could not possibly be of any help in building the classifiers.

As for the classifier learning phase, in order to obtain more reliable conclusions we have performed our experiments alternately with two different state-of-the-art learning algorithms, ADABOOST.MH^{KR} and SVMLIGHT.

ADABOOST.MH^{KR} is a "boosting" algorithm proposed in [39] and subsequently improved in [25], which modifies and improves upon the ADABOOST.MH^{R} algorithm described in [32]. 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 $k-1$ previously trained classifiers.

SVMLIGHT (version 3.5)[6] is instead a support vector machine (SVM) learner [18]. SVMs attempt to learn a hyperplane in n-dimensional space (where $n = |\mathcal{D}|$ for the DOR and $n = |\mathcal{T}|$for the TCOR) that separates the positive training examples from the negative ones with the maximum possible margin, i.e. in such a way that the minimal distance between the hyperplane and a training example is maximum. Results in computational learning theory indicate that this tends to minimize the generalization error, i.e. the error of the resulting classifier on yet unseen examples. We have simply opted for the default parameter setting of SVMLIGHT; in particular, this means that a linear kernel has been used.

As the evaluation metric we have chosen the widely used F_1 function, in both its *microaveraged* and *macroaveraged* variants. F_1 is the harmonic mean $F_1 = \frac{2\pi\rho}{\pi+\rho}$ of precision (π) and recall (ρ), which may be viewed as the degree of soundness and the degree of completeness of a classifier, respectively [38]. In the microaveraged variant categories count proportionally to the number of their positive test examples, while in the macroaveraged variant all categories

[5] For reasons of space we avoid to include precision and recall values, and only report F_1 values; a complete listing of all precision and recall values for the experiments reported in this paper is included as an "online appendix" of this paper, which can be downloaded from `http://www.isti.cnr.it/People/F.Sebastiani/DSL-experiments.xls`

[6] `http://svmlight.joachims.org/`

are attributed the same importance.

3.1.2 The results

The results of our experiments are reported in Figure 1[7]. It can be seen that both ADABOOST.MH^{KR} and SVMLIGHT perform significantly better with the TCOR than with the DOR; in terms of microaveraged F_1, ADABOOST.MH^{KR} performs up to +25.6% better (for $x = 60$) with the TCOR, and the differential for SVMLIGHT is even higher (+49.7%). The differential is higher for microaveraged F_1 than for macroaveraged F_1, which indicates that the TCOR is especially suited to working with categories with large numbers of positive examples.

In a further set of experiments we have compared the DOR and the TCOR by using sentences (instead of documents) as "reference contexts" (see Section 2.3). That sentences might perform better than documents when used as reference contexts is plausible, since it is plausible that the longer the context, the less significant the co-occurrence of two terms is in indicating that they belong to the same category.

We have run this set of experiments by segmenting into sentences (i.e. using the full stop as separator) each of the documents considered in the previous experiments, and by considering each of the resulting 714,352 sentences as a reference context[8]. The results (reported in Figure 2) confirm that the TCOR is consistently superior to the DOR, for all values of x, both for micro- and for macro-averaged F_1. The differential in performance is even more dramatic than when using documents as reference contexts, and tends to increase with the value of x, to the point that for $x = 60$ and microaveraged F_1 the TCOR is 100% better than the DOR with ADABOOST.MH^{KR} and 421% better with SVMLIGHT. Interestingly, the DOR performs much better with documents as reference contexts, while the TCOR performs slightly better when the reference contexts used are sentences[9].

3.2 Term clustering experiments

In order to find further experimental evidence for our comparison between the DOR and the TCOR, we have performed a set of term clustering experiments, i.e. experiments in which we try to partition a set Θ of terms into m semantically coherent groups of terms. This task thus corresponds to discovering a latent, hidden structure within this set of terms.

The typical approach to term clustering consists in the application of a clustering algorithm [17] to the representations of the terms to be clustered. For this to be possible, a similarity measure must be defined on these representations, so as to allow the algorithm to generate a cluster structure which maximizes the similarity between two terms belonging to the same cluster and minimizes the similarity between two terms belonging to different clusters.

For these experiments we have used CLUTO[10], an off-the-shelf software package for clustering high-dimensional datasets. We have simply opted for the default parameter setting of CLUTO; this means using a partitional clustering algorithm (whereby the desired m-way clustering solution is computed by performing a sequence of $m - 1$ repeated bisections) and a cosine similarity function.

3.2.1 The experimental setting

Following a consolidated practice, we here measure the effectiveness of a clustering system by the degree to which it is able to "correctly" re-classify a set Θ of pre-classified terms into exactly the same categories without knowing the original category assignment. In other words, given a set $C = \{c_1, \ldots, c_m\}$ of categories, and a set Θ of terms preclassified under C, the "ideal" term clustering algorithm is the one that, when asked to cluster Θ into m groups, produces a grouping $C' = \{c'_1, \ldots, c'_m\}$ such that, for each term $t_j \in \Theta$, $t_j \in c_i$ if and only if $t_j \in c'_i$. The original labelling is thus viewed as the latent, hidden structure that the clustering system must discover.

Following [44, page 110], the measure we use is *normalized mutual information* (NMI), i.e.

$$NMI(C, C') = \frac{2}{|\Theta|} \sum_{c \in C} \sum_{c' \in C'} P(c, c') \cdot \log_{|C| \cdot |C'|} \frac{P(c, c')}{P(c) \cdot P(c')}$$

where $P(c)$ represents the probability that a randomly selected term t_j belongs to c, and $P(c, c')$ represents the probability that a randomly selected term t_j belongs to both c and c'[11].

For our experiments we thus need, again, two datasets: a set Θ of terms preclassified according to a set $C = \{c_1, \ldots, c_m\}$ of categories, and a set Δ of documents that provide the "implicit" representations for the terms. As for Θ, we have again chosen WordNetDomains(42). However, since CLUTO is a clustering system that returns *non-overlapping clusters* (i.e. each $t_j \in \Theta$ belongs to one and only one of the generated clusters), we only consider the *single-category fraction* of WordNetDomains(42), i.e. the set of WordNetDomains(42) terms that belong to one and one category only. There are 9,749 such single-category terms in WordNetDomains(42). As for the corpus Δ, we have again considered, as in our term categorization experiments, the November 1996 fragment of RCV1[12]. The feature vectors used to represent the

[7] The reader might note that F_1 scores are much worse than F_1 scores that have been reported in the literature, for the same test collection, on the related task of *text* categorization. The interested reader may consult [2] for a detailed discussion of this point, which is not within the scope of this paper.

[8] We did not run experiments using paragraphs instead of sentences (i.e. using a carriage return as the separator) since most paragraphs in RCV1 documents consist of single sentences.

[9] However, note that for a "fair" comparison between sentences and documents as reference contexts, the ratio between training terms and reference contexts should be the same in the two runs being compared, in order to prevent overfitting. This is not the case here since we have approximately 12 times many more sentences than documents. See [2] for more discussion on this.

[10] `http://www-users.cs.umn.edu/~karypis/cluto/`

[11] In all our term clustering experiments we have also computed the alternative measures of *purity* and *entropy* [44, pages 107–108]; we do not report them explicitly since they plainly confirm the observations that can be drawn from the MI figures.

[12] However, note that after removing the terms with multiple categories, one category (ASTROLOGY) becomes empty. Further categories become empty for values of $x > 1$; for instance, when $x = 10$ the number of nonempty categories decreases to 40, and to 39 for $x = 30$. Note that in these cases the number of clusters which CLUTO was asked to produce was reduced accordingly.

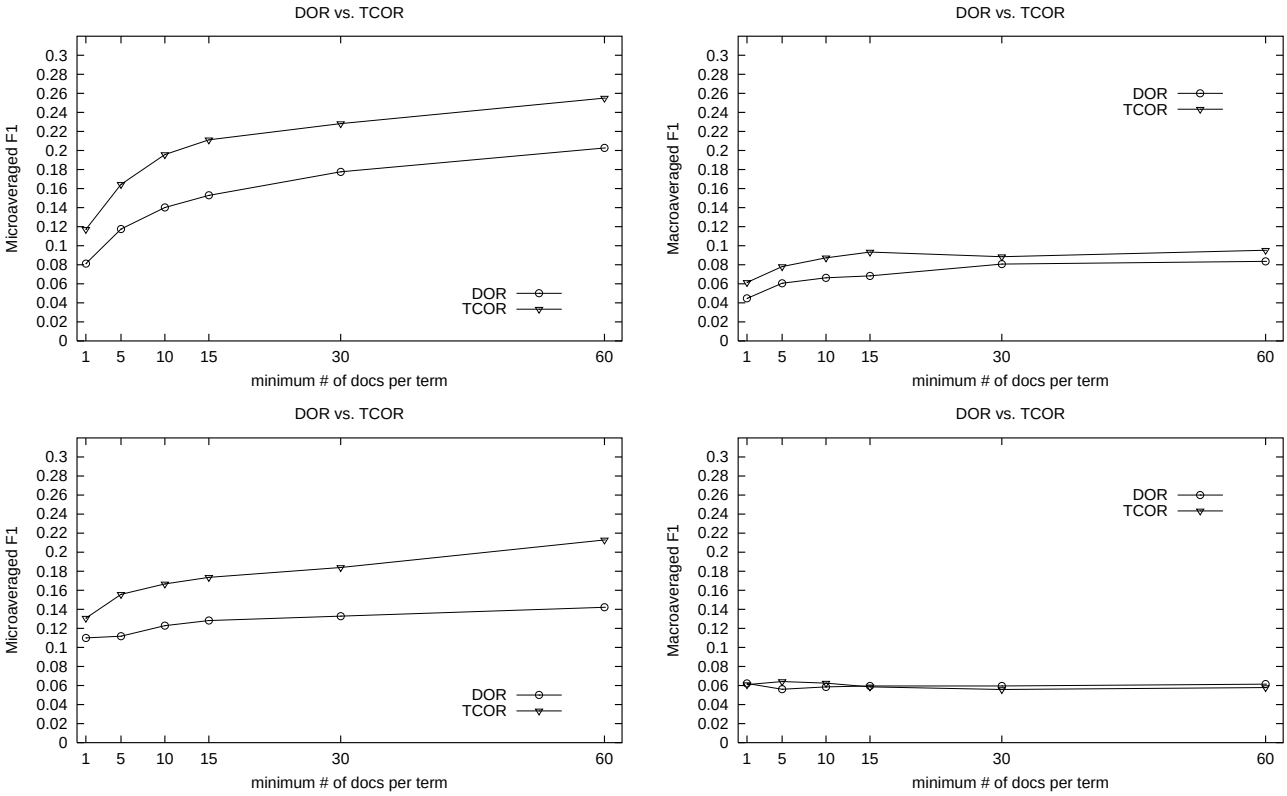

Figure 1: Comparison of the results obtained with AdaBoost.MH^{KR} (top) and SVMlight (bottom) with the DOR and TCOR. Plots report micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) as a function of x, which represents the minimal number of documents in which training and test terms must occur in order to be taken into consideration.

terms were exactly the same as those used in the supervised term categorization experiments of the previous sections. As in the term categorization experiments, we have repeated each experiment several times by considering only terms occurring in at least x documents, for each value of $x \in \{1, 5, 10, 15, 30, 60\}$.

3.2.2 The results

The results of the experiments (which are illustrated in Figure 3) substantially confirm the results of our previous term categorization experiments, i.e. they confirm that the TCOR performs better than the DOR, although the tendency is less consistent (when documents are used as reference contexts, for $x \geq 10$ the DOR outperforms the TCOR) and the improvement is quantitatively less marked.

4. WHY DOES THE TCOR PERFORM BETTER?

Why does the TCOR perform better than the DOR? We think the reason may be due to the fact that the TCOR captures some phenomena related to semantic similarity better than the DOR. One example of this is the behaviour of the two representations when dealing with perfect synonymy. To see this, observe that one characteristics of a good representation for terms is that it tends to produce similar representations for semantically similar terms, and identical representations for semantically identical terms (i.e. perfect synonyms). Now, two perfect synonyms t_1 and t_2 may be represented by fairly dissimilar vectors according to the DOR, since an author might typically use, throughout a given document, either the one or the other term, but not both, for better consistency. If *all* authors of the documents in $\mathcal{D}$ had used this policy, t_1 and t_2 would never co-occur in the same document, and the degree of similarity of the corresponding vectors would be (according to standard vector-based similarity functions such as dot product or cosine) zero. This needs not be a problem with the TCOR, since the two terms need not frequently co-occur *with each other* in 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 (e.g. it is likely that `jail` and `prison` will both frequently co-occur with `prisoner` and that they will seldom co-occur with each other).

There is a further explanation of the substantial difference in performance between the DOR and the TCOR, which is based on information-theoretic considerations. This explanation has to do with the fact that, as we have found, the discriminative power of TCOR features is much higher than the discriminative power of DOR features. In order to see this we have computed, for our two representations, and using documents (not sentences) as the reference context, the *mutual information* (MI) function ("globalized" by taking

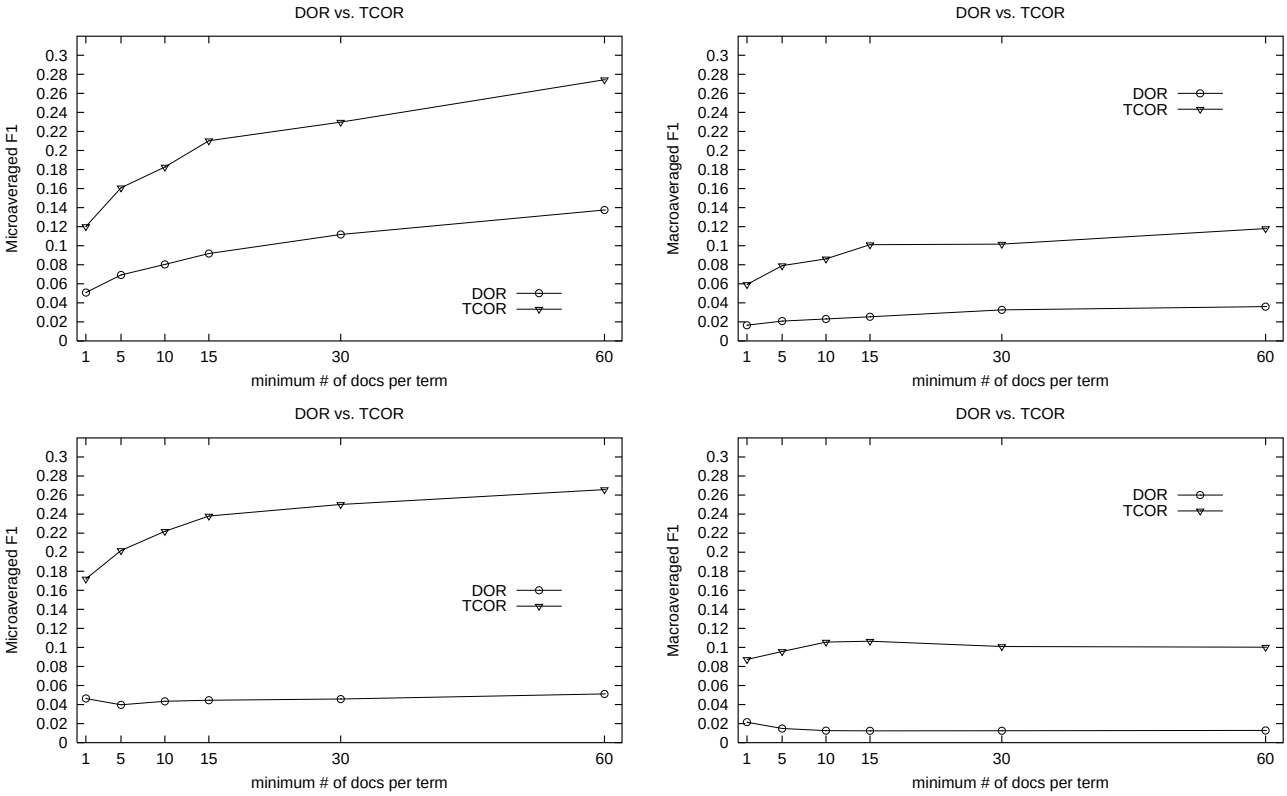

Figure 2: Comparison of the results obtained with AdaBoost.MH^{KR} (top) and SVMlight (bottom) with the DOR and TCOR using sentences as reference contexts.

the maximum of its "local", category-specific values), defined as[13]

$$MI(f_k) = \max_{i=1}^{m} MI(f_k, c_i) \quad (7)$$
$$= \max_{i=1}^{m} \sum_{c \in \{c_i, \overline{c}_i\}} \sum_{f \in \{f_k, \overline{f}_k\}} P(f,c) \cdot \log \frac{P(f,c)}{P(f) \cdot P(c)}$$

for the features f_k that score highest in terms of such a function. In categorization applications MI serves the purpose of measuring the discriminative power of a feature with respect to a set of classes $C = \{c_1, \ldots, c_m\}$, i.e. of evaluating the expected quality of the contribution that a feature will give to the categorization task.

The results, which are computed on the *entire* RCV1 corpus of documents, are plotted in Figure 4. From the figure we can see that, on average, a feature of the TCOR has a value much higher than the feature of the same rank in the DOR. This confirms our hypothesis that the features used in the TCOR (i.e. the terms that co-occur with t_j) are *inherently better features* than the features used in the DOR (i.e. the documents in which t_j occurs). This proves that the better performance of the TCOR with respect to the DOR, that we have observed in our experiments, is not contingently due to the use of AdaBoost.MH^{KR} and SVMlight as categorization algorithms, (resp. of Cluto as clustering algorithm), but could be expected even if different categorization (resp. clustering) algorithms were employed.

[13]Mutual information is a function from information theory which is also known as *information gain*, and is sometimes given in the equivalent form $MI(f_k, c_i) = H(c_i) - H(c_i|t_k)$, where $H(X)$ is the *entropy* of X and $H(X|Y)$ is the *conditional entropy* of Y given X [5, page 19]. The NMI function introduced in Section 3.2.1 is a normalized version of MI, which is meant to eliminate the bias due to the different entropies of C and C'.

5. RELATED WORK

As previously mentioned throughout the paper, the use (either implicit or explicit) of statistics on term co-occurrence for determining vectorial representations of the semantics of terms has a long history, in computational linguistics [13, 21, 26, 30, 35, 36], information retrieval [33, 37, 40, 41], and even psycholinguistics [22]. While applications of one or the other approach to term representation have been many, comparisons among different representation styles have been few. All such comparisons have been, to the best of our knowlegde, "internal" to either the DOR or the TCOR.

For instance, Sahlgren [30], who uses a variant of the TCOR based on the notion of "random indexing", compares representations that make use or do not make use, respectively, of syntactic information; however, both representations use terms as features and use a sliding fixed-size window as the context in which co-occurrence is deemed significant. Sahlgren also mentions the existence of the DOR, but seems to assume that the substantial difference between the DOR and the TCOR is that the DOR necessarily uses the entire document as the reference context while the TCOR

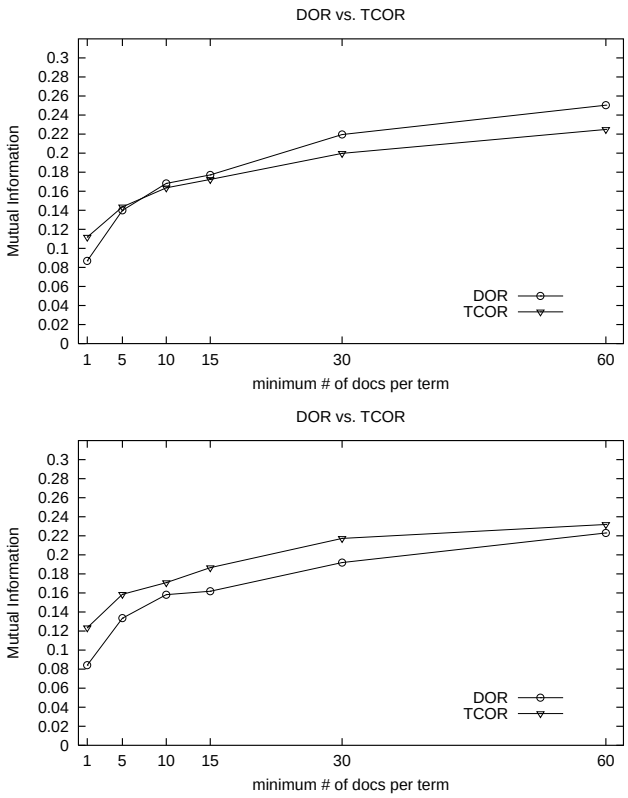

Figure 3: Comparison of the results obtained with Cluto, using full documents (top) or sentences (bottom) as reference contexts, with the DOR and TCOR.

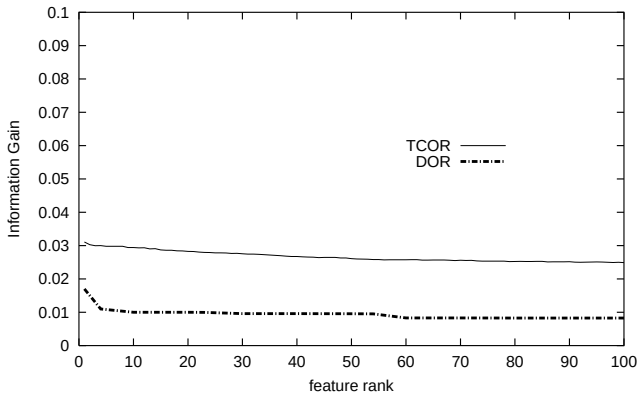

Figure 4: Comparison of the absolute values of the mutual information function for features in the DOR and in the TCOR, as a function of the rank of such features. The results are reported for the 100 top-scoring features only.

may use (and typically uses) shorter portions of the text, such as a sentence or a sliding fixed-sized window. As our experiments show, this is not the case, and the two representations differ in effectiveness also when the entire document is used as the reference context.

The issue of deciding whether, in the TCOR, shorter or longer reference contexts should be used, has been the subject of several papers. Most authors in CL (e.g. [22, 30, 37]) seem to lean towards the use of shorter contexts, based on the hypothesis that the semantically significant context of a term is its immediate vicinity. Sahlgren [30] suggests that shorter, "local" contexts tend to identify a stronger notion of similarity (somehow related to the standard notion of synonymy as from lexicography), while longer, "global" contexts tend to identify a weaker notion (somehow related to the notion of thematic relatedness).

In sum, comparisons between the DOR and the TCOR, either of an experimental or of a speculative nature, have not been performed to date to the best of our knowledge.

6. CONCLUSION

We have presented an experimental comparison of two alternative distributional representations for terms, the document occurrence representation (DOR) and the term co-occurrence representation (TCOR), that have been independently developed by the information retrieval community and by the computational linguistics community, respectively, and whose relative merits had never been assessed so far. We have performed this comparison in terms of a term categorization task and a term clustering task, both run on a large corpus of 42 domain-specific lexicons and on a medium-sized corpus of more than 65,000 documents. Our results have shown that the TCOR almost always outperforms the DOR, sometimes even spectacularly (up to 421% improvement when using sentences as reference contexts in term categorization). We have given an intuitive argument for this, based on how the two representations deal with synonymy. We have further provided an information-theoretic argument that confirms this intuition, based on assessing the discriminative power of the two types of features by means of the mutual information function.

These results are of potential interest to anyone working in content management applications (such as term clustering, word sense disambiguation, or automated thesaurus generation) that require terms to be explicitly represented by means of distributional representations, and may come as a surprise especially to people who, in the IR tradition, had been working with the DOR, maybe ignoring the existence of an alternative representation style.

In the future we plan to confirm the results of this investigation by testing different, more sophisticated weighting functions (e.g. Robertson's BM25 function), and by testing them on term management tasks other than term categorization and clustering.

Acknowledgments

We wish to thank Thorsten Joachims and George Karypis for making available the SVMlight and Cluto packages, respectively. Thanks also to Ido Dagan and Alessandro Lenci for several pointers to the computational linguistics literature on term representation, and to Joydeep Ghosh and Alexander Strehl for useful discussions.

7. REFERENCES

[1] H. Avancini, A. Lavelli, B. Magnini, F. Sebastiani, and R. Zanoli. Expanding domain-specific lexicons by

term categorization. In *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing*, pages 793–797, Melbourne, US, 2003. ACM Press, New York, US. An extended version appears as [2].

[2] H. Avancini, A. Lavelli, F. Sebastiani, and R. Zanoli. Automatic expansion of domain-specific lexicons by term categorization. 2004. Submitted for publication. Available at `http://www.isti.cnr.it/People/F.Sebastiani/Publications/TSLP04.pdf`.

[3] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[4] H. Chen, T. Yim, D. Fye, and B. R. Schatz. Automatic thesaurus generation for an electronic community system. *Journal of the American Society for Information Science*, 46(3):175–193, 1995.

[5] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, New York, US, 1991.

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

[7] C. J. Crouch and B. Yang. 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, 1992.

[8] I. Dagan. Contextual word similarity. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*, chapter 19, pages 459–476. Marcel Dekker Inc, New York, NY, 2000.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[10] S. P. Finch. *Finding Structure in Language*. PhD thesis, University of Edinburgh, Edinburgh, UK, 1993.

[11] J. R. Firth. A synopsis of linguistic theory 1930-1955. In F. Palmer, editor, *Selected papers of J.R. Firth*. Longman, Harlow, UK, 1968.

[12] W. A. Gale, K. W. Church, and D. Yarowsky. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439, 1993.

[13] S. I. Gallant. A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation*, 3(3):293–309, 1991.

[14] J. Galliers and K. Spärck-Jones. *Evaluating Natural Language Processing Systems*. Springer, Berlin, DE, 1996.

[15] G. Grefenstette. *Explorations in automatic thesaurus discovery*. Kluwer Academic Publishers, Dordrecht, NL, 1994.

[16] Z. Harris. *Mathematical structures of language*. John Wiley & Sons, New York, US, 1968.

[17] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering. *ACM Computing Surveys*, 31(3):264–323, 1998.

[18] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. J. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 11, pages 169–184. The MIT Press, Cambridge, US, 1999.

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

[20] D. D. Lewis and W. B. Croft. Term clustering of syntactic phrases. In *Proceedings of SIGIR-90, 13th ACM International Conference on Research and Development in Information Retrieval*, pages 385–404, Bruxelles, BE, 1990.

[21] D. Lin. Automatic retrieval and clustering of similar words. In *Proceedings of ACL-98, 36th Annual Meeting of the Association for Computational Linguistics*, pages 768–774, Montreal, CA, 1998.

[22] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers*, 28(2):203–208, 1996.

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

[24] J. Minker, G. A. Wilson, and B. H. Zimmermann. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, 8:329–348, 1972.

[25] P. Nardiello, F. Sebastiani, and A. Sperduti. Discretizing continuous attributes in AdaBoost for text categorization. In F. Sebastiani, editor, *Proceedings of ECIR-03, 25th European Conference on Information Retrieval*, pages 320–334, Pisa, IT, 2003. Springer Verlag.

[26] P. Pantel and D. Lin. Automatically discovering word senses. In *Proceedings of HLT-03, 3rd International Conference on Human Language Technology*, pages 21–22, Edmonton, CA, 2003.

[27] F. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *Proceedings of ACL-93, 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, US, 1993.

[28] Y. Qiu and H.-P. Frei. Concept-based query expansion. In *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 160–169, Pittsburgh, US, 1993.

[29] T. Rose, M. Stevenson, and M. Whitehead. The Reuters Corpus Volume 1 – from yesterday's news to tomorrow's language resources. In *Proceedings of LREC-02, 3rd International Conference on Language Resources and Evaluation*, pages 827–832, Las Palmas, ES, 2002.

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

[31] G. Salton. Experiments in automatic thesaurus

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

[32] R. E. Schapire and Y. Singer. BoosTexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[33] P. Schäuble and D. Knaus. The various roles of information structures. In O. Opitz, B. Lausen, and R. Klar, editors, *Proceedings of the 16th Annual Conference of the Gesellschaft für Klassifikation*, pages 282–290, Dortmund, DE, 1992. Published by Springer Verlag, Heidelberg, DE, 1993.

[34] H. Schmid. 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, 1994.

[35] H. Schütze. Dimensions of meaning. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pages 787–796, Minneapolis, US, 1992.

[36] H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124, 1998.

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

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

[39] F. Sebastiani, A. Sperduti, and N. Valdambrini. An improved boosting algorithm and its application to automated text categorization. In A. Agah, J. Callan, and E. Rundensteiner, editors, *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 78–85, McLean, US, 2000. ACM Press, New York, US.

[40] P. Sheridan and J.-P. Ballerini. 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, 1996.

[41] P. Sheridan, M. Braschler, and P. Schäuble. Cross-language information retrieval in a multi-lingual legal domain. In C. Peters and C. Thanos, editors, *Proceedings of ECDL-97, 1st European Conference on Research and Advanced Technology for Digital Libraries*, pages 253–268, Pisa, IT, 1997. Lecture Notes in Computer Science, number 1324, Springer Verlag, Heidelberg, DE.

[42] K. Spärck Jones. *Automatic keyword classification for information retrieval*. Butterworths, London, UK, 1971.

[43] K. Spärck Jones. Collection properties influencing automatic term classification performance. *Information Storage and Retrieval*, 9:499–513, 1973.

[44] A. Strehl. *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, University of Texas, Austin, US, 2002.