Unsupervised Domain Relevance Estimation for Word Sense Disambiguation

Alfio Gliozzo and **Bernardo Magnini** and **Carlo Strapparava**
ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica, I-38050 Trento, ITALY
{gliozzo, magnini, strappa}@itc.it

Abstract

This paper presents *Domain Relevance Estimation* (*DRE*), a fully unsupervised text categorization technique based on the statistical estimation of the relevance of a text with respect to a certain category. We use a pre-defined set of categories (we call them *domains*) which have been previously associated to WORDNET word senses. Given a certain domain, DRE distinguishes between relevant and non-relevant texts by means of a Gaussian Mixture model that describes the frequency distribution of domain words inside a large-scale corpus. Then, an Expectation Maximization algorithm computes the parameters that maximize the likelihood of the model on the empirical data.

The correct identification of the domain of the text is a crucial point for Domain Driven Disambiguation, an unsupervised Word Sense Disambiguation (WSD) methodology that makes use of only domain information. Therefore, DRE has been exploited and evaluated in the context of a WSD task. Results are comparable to those of state-of-the-art unsupervised WSD systems and show that DRE provides an important contribution.

1 Introduction

A fundamental issue in text processing and understanding is the ability to detect the topic (i.e. the domain) of a text or of a portion of it. Indeed, domain detection allows a number of useful simplifications in text processing applications, such as, for instance, in Word Sense Disambiguation (WSD).

In this paper we introduce Domain Relevance Estimation (DRE) a fully unsupervised technique for domain detection. Roughly speaking, DRE can be viewed as a text categorization (TC) problem (Sebastiani, 2002), even if we do not approach the problem in the standard supervised setting requiring category labeled training data. In fact, recently, unsupervised approaches to TC have received more and more attention in the literature (see for example (Ko and Seo, 2000).

We assume a pre-defined set of categories, each defined by means of a list of related terms. We call such categories *domains* and we consider them as a set of general topics (e.g. SPORT, MEDICINE, POLITICS) that cover the main disciplines and areas of human activity. For each domain, the list of related words is extracted from WORDNET DOMAINS (Magnini and Cavaglià, 2000), an extension of WORDNET in which synsets are annotated with domain labels. We have identified about 40 domains (out of 200 present in WORDNET DOMAINS) and we will use them for experiments throughout the paper (see Table 1).

DRE focuses on the problem of estimating a degree of relatedness of a certain text with respect to the domains in WORDNET DOMAINS.

The basic idea underlying DRE is to combine the knowledge in WORDNET DOMAINS and a probabilistic framework which makes use of a large-scale corpus to induce domain frequency distributions. Specifically, given a certain domain, DRE considers frequency scores for both relevant and non-relevant texts (i.e. texts which introduce noise) and represent them by means of a Gaussian Mixture model. Then, an Expectation Maximization algorithm computes the parameters that maximize the likelihood of the empirical data.

DRE methodology originated from the effort to improve the performance of Domain Driven Disambiguation (DDD) system (Magnini et al., 2002). DDD is an unsupervised WSD methodology that makes use of only domain information. DDD assignes the right sense of a word in its context comparing the domain of the context to the domain of each sense of the word. This methodology exploits WORDNET DOMAINS information to estimate both

Domain	#Syn	Domain	#Syn	Domain	#Syn
Factotum	36820	Biology	21281	Earth	4637
Psychology	3405	Architecture	3394	Medicine	3271
Economy	3039	Alimentation	2998	Administration	2975
Chemistry	2472	Transport	2443	Art	2365
Physics	2225	Sport	2105	Religion	2055
Linguistics	1771	Military	1491	Law	1340
History	1264	Industry	1103	Politics	1033
Play	1009	Anthropology	963	Fashion	937
Mathematics	861	Literature	822	Engineering	746
Sociology	679	Commerce	637	Pedagogy	612
Publishing	532	Tourism	511	Computer_Science	509
Telecommunication	493	Astronomy	477	Philosophy	381
Agriculture	334	Sexuality	272	Body_Care	185
Artisanship	149	Archaeology	141	Veterinary	92
Astrology	90				

Table 1: Domain distribution over WORDNET synsets.

the domain of the textual context and the domain of the senses of the word to disambiguate. The former operation is intrinsically an unsupervised TC task, and the category set used has to be the same used for representing the domain of word senses.

Since DRE makes use of a fixed set of target categories (i.e. domains) and since a document collection annotated with such categories is not available, evaluating the performance of the approach is a problem in itself. We have decided to perform an indirect evaluation using the DDD system, where unsupervised TC plays a crucial role.

The paper is structured as follows. Section 2 introduces WORDNET DOMAINS, the lexical resource that provides the underlying knowledge to the DRE technique. In Section 3 the problem of estimating domain relevance for a text is introduced. In particular, Section 4 briefly sketchs the WSD system used for evaluation. Finally, Section 5 describes a number of evaluation experiments we have carried out.

2 Domains, WORDNET and Texts

DRE heavily relies on domain information as its main knowledge source. Domains show interesting properties both from a lexical and a textual point of view. Among these properties there are: (i) lexical coherence, since part of the lexicon of a text is composed of words belonging to the same domain; (ii) polysemy reduction, because the potential ambiguity of terms is sensibly lower if the domain of the text is specified; and (iii) lexical identifiability of text's domain, because it is always possible to assign one or more domains to a given text by considering term distributions in a bag-of-words approach. Experimental evidences of these properties are reported in (Magnini et al., 2002).

In this section we describe WORDNET DOMAINS[1] (Magnini and Cavaglià, 2000), a lexical resource that attempts a systematization of relevant aspects in domain organization and representation. WORDNET DOMAINS is an extension of WORDNET (version 1.6) (Fellbaum, 1998), in which each synset is annotated with one or more domain labels, selected from a hierarchically organized set of about two hundred labels. In particular, issues concerning the "completeness" of the domain set, the "balancing" among domains and the "granularity" of domain distinctions, have been addressed. The domain set used in WORDNET DOMAINS has been extracted from the Dewey Decimal Classification (Comaroni et al., 1989), and a mapping between the two taxonomies has been computed in order to ensure completeness. Table 2 shows how the senses for a word (i.e. the noun *bank*) have been associated to domain label; the last column reports the number of occurrences of each sense in Semcor[2].

Domain labeling is complementary to information already present in WORDNET. First of all, a domain may include synsets of different syntactic categories: for instance MEDICINE groups together senses from nouns, such as `doctor#1` and `hospital#1`, and from verbs, such as `operate#7`. Second, a domain may include senses from different WORDNET sub-hierarchies (i.e. deriving from different "unique beginners" or from different "lexicographer files"). For example, SPORT contains senses such as `athlete#1`, deriving from `life_form#1`, `game_equipment#1` from `physical_object#1`, `sport#1`

[1] WORDNET DOMAINS is freely available at `http://wndomains.itc.it`

[2] SemCor is a portion of the Brown corpus in which words are annotated with WORDNET senses.

Sense	Synset and Gloss	Domains	Semcor frequencies
#1	depository financial institution, bank, banking concern, banking company (a financial institution...)	ECONOMY	20
#2	bank (sloping land...)	GEOGRAPHY, GEOLOGY	14
#3	bank (a supply or stock held in reserve...)	ECONOMY	-
#4	bank, bank building (a building...)	ARCHITECTURE, ECONOMY	-
#5	bank (an arrangement of similar objects...)	FACTOTUM	1
#6	savings bank, coin bank, money box, bank (a container...)	ECONOMY	-
#7	bank (a long ridge or pile...)	GEOGRAPHY, GEOLOGY	2
#8	bank (the funds held by a gambling house...)	ECONOMY, PLAY	
#9	bank, cant, camber (a slope in the turn of a road...)	ARCHITECTURE	-
#10	bank (a flight maneuver...)	TRANSPORT	-

Table 2: WORDNET senses and domains for the word "bank".

from `act#2`, and `playing_field#1` from `location#1`.

Domains may group senses of the same word into thematic clusters, which has the important side-effect of reducing the level of ambiguity when we are disambiguating to a domain. Table 2 shows an example. The word "bank" has ten different senses in WORDNET 1.6: three of them (i.e. `bank#1`, `bank#3` and `bank#6`) can be grouped under the ECONOMY domain, while `bank#2` and `bank#7` both belong to GEOGRAPHY and GEOLOGY. Grouping related senses is an emerging topic in WSD (see, for instance (Palmer et al., 2001)).

Finally, there are WORDNET synsets that do not belong to a specific domain, but rather appear in texts associated with any domain. For this reason, a FACTOTUM label has been created that basically includes *generic* synsets, which appear frequently in different contexts. Thus the FACTOTUM domain can be thought of as a "placeholder" for all other domains.

3 Domain Relevance Estimation for Texts

The basic idea of domain relevance estimation for texts is to exploit lexical coherence inside texts. From the domain point of view lexical coherence is equivalent to domain coherence, i.e. the fact that a great part of the lexicon inside a text belongs to the same domain.

From this observation follows that a simple heuristic to approach this problem is counting the occurrences of domain words for every domain inside the text: the higher the percentage of domain words for a certain domain, the more relevant the domain will be for the text. In order to perform this operation the WORDNET DOMAINS information is exploited, and each word is assigned a weighted list of domains considering the domain annotation of its synsets. In addition, we would like to estimate the domain of the text locally. Local estimation of domain relevance is very important in order to take into account domain shifts inside the text. The methodology used to estimate domain frequency is described in subsection 3.1.

Unfortunately the simple local frequency count is not a good domain relevance measure for several reasons. The most significant one is that very frequent words have, in general, many senses belonging to different domains. When words are used in texts, ambiguity tends to disappear, but it is not possible to assume knowing their actual sense (i.e. the sense in which they are used in the context) in advance, especially in a WSD framework. The simple frequency count is then inadequate for relevance estimation: irrelevant senses of ambiguous words contribute to augment the final score of irrelevant domains, introducing noise. The level of noise is different for different domains because of their different sizes and possible differences in the ambiguity level of their vocabularies.

In subsection 3.2 we propose a solution for that problem, namely the Gaussian Mixture (GM) approach. This constitutes an unsupervised way to estimate how to differentiate relevant domain information in texts from noise, because it requires only a large-scale corpus to estimate parameters in an Expectation Maximization (EM) framework. Using the estimated parameters it is possible to describe the distributions of both relevant and non-relevant texts, converting the DRE problem into the problem of estimating the probability of each domain given its frequency score in the text, in analogy to the bayesian classification framework. Details about the EM algorithm for GM model are provided in subsection 3.3.

3.1 Domain Frequency Score

Let $t \in \mathcal{T}$, be a text in a corpus $\mathcal{T}$ composed by a list of words $w_1^t, \ldots, w_q^t$. Let $\mathcal{D} = \{D_1, D_2, ..., D_d\}$ be

the set of domains used. For each domain D_k the domain "frequency" score is computed in a window of c words around w_j^t. The domain frequency score is defined by formula (1).

$$F(D_k, t, j) = \sum_{i=j-c}^{j+c} R_{word}(D_k, w_i^t) G(i, j, (\frac{c}{2})^2) \quad (1)$$

where the weight factor $G(x, \mu, \sigma^2)$ is the density of the normal distribution with mean μ and standard deviation σ at point x and $R_{word}(D, w)$ is a function that return the relevance of a domain D for a word w (see formula 3). In the rest of the paper we use the notation $F(D_k, t)$ to refer to $F(D_k, t, m)$, where m is the integer part of $q/2$ (i.e. the "central" point of the text - q is the text length).

Here below we see that the information contained in WORDNET DOMAINS can be used to estimate $R_{word}(D_k, w)$, i.e. domain relevance for the word w, which is derived from the domain relevance of the synsets in which w appears.

As far as synsets are concerned, domain information is represented by the function $Dom : S \Rightarrow P(D)$[3] that returns, for each synset $s \in S$, where S is the set of synsets in WORDNET DOMAINS, the set of the domains associated to it. Formula (2) defines the domain relevance estimation function (remember that d is the cardinality of $\mathcal{D}$):

$$R_{syn}(D, s) = \begin{cases} 1/|Dom(s)| & : \texttt{if } D \in Dom(s) \\ 1/d & : \texttt{if } Dom(s) = \{\text{FACTOTUM}\} \\ 0 & : \texttt{otherwise} \end{cases} \quad (2)$$

Intuitively, $R_{syn}(D, s)$ can be perceived as an estimated prior for the probability of the domain given the concept, as expressed by the WORDNET DOMAINS annotation. Under these settings FACTOTUM (generic) concepts have uniform and low relevance values for each domain while domain concepts have high relevance values for a particular domain.

The definition of domain relevance for a word is derived directly from the one given for concepts. Intuitively a domain D is relevant for a word w if D is relevant for one or more senses c of w. More formally let $V = \{w_1, w_2, ...w_{|V|}\}$ be the vocabulary, let $senses(w) = \{s | s \in S, s \text{ is a sense of } w\}$ (e.g. any synset in WORDNET containing the word w). The domain relevance function for a word $R : \mathcal{D} \times V \Rightarrow [0, 1]$ is defined as follows:

$$R_{word}(D_i, w) = \frac{1}{|senses(w)|} \sum_{s \in senses(w)} R_{syn}(D_i, s) \quad (3)$$

[3] $P(D)$ denotes the power set of D

3.2 The Gaussian Mixture Algorithm

As explained at the beginning of this section, the simple local frequency count expressed by formula (1) is not a good domain relevance measure.

In order to discriminate between noise and relevant information, a supervised framework is typically used and significance levels for frequency counts are estimated from labeled training data. Unfortunately this is not our case, since no domain labeled text corpora are available. In this section we propose a solution for that problem, namely the Gaussian Mixture approach, that constitutes an unsupervised way to estimate how to differentiate relevant domain information in texts from noise. The Gaussian Mixture approach consists of a parameter estimation technique based on statistics of word distribution in a large-scale corpus.

The underlying assumption of the Gaussian Mixture approach is that frequency scores for a certain domain are obtained from an underlying mixture of relevant and non-relevant texts, and that the scores for relevant texts are significantly higher than scores obtained for the non-relevant ones. In the corpus these scores are distributed according to two distinct components. The domain frequency distribution which corresponds to relevant texts has the higher value expectation, while the one pertaining to non relevant texts has the lower expectation. Figure 1 describes the probability density function (PDF) for domain frequency scores of the SPORT domain estimated on the BNC corpus[4] (BNC-Consortium, 2000) using formula (1). The "empirical" PDF, describing the distribution of frequency scores evaluated on the corpus, is represented by the continuous line.

From the graph it is possible to see that the empirical PDF can be decomposed into the sum of two distributions, D = SPORT and $\overline{D}$ = "non-SPORT". Most of the probability is concentrated on the left, describing the distribution for the majority of non relevant texts; the smaller distribution on the right is assumed to be the distribution of frequency scores for the minority of relevant texts.

Thus, the distribution on the left describes the noise present in frequency estimation counts, which is produced by the impact of polysemous words and of occasional occurrences of terms belonging to SPORT in non-relevant texts. The goal of the technique is to estimate parameters describing the distribution of the noise along texts, in order to as-

[4] The British National Corpus is a very large (over 100 million words) corpus of modern English, both spoken and written.

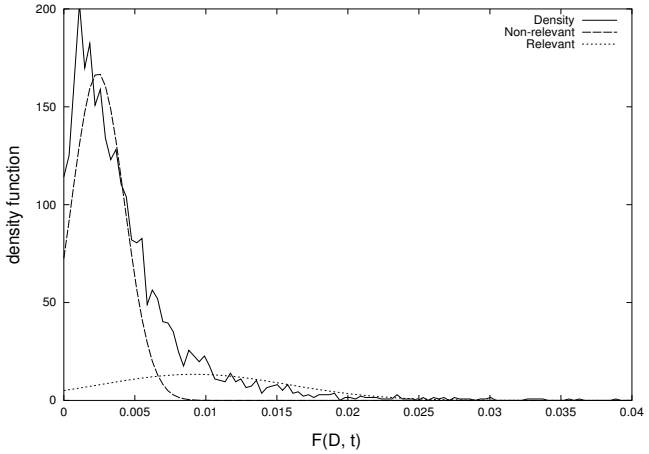

Figure 1: Gaussian mixture for D = SPORT

sociate high relevance values only to relevant frequency scores (i.e. frequency scores that are not related to noise). It is reasonable to assume that such noise is normally distributed because it can be described by a binomial distribution in which the probability of the positive event is very low and the number of events is very high. On the other hand, the distribution on the right is the one describing typical frequency values for relevant texts. This distribution is also assumed to be normal.

A probabilistic interpretation permits the evaluation of the relevance value $R(D,t,j)$ of a certain domain D for a new text t in a position j only by considering the domain frequency $F(D,t,j)$. The relevance value is defined as the conditional probability $P(D|F(D,t,j))$. Using Bayes theorem we estimate this probability by equation (4).

$$R(D,t,j) \quad = \quad P(D|F(D,t,j)) = \tag{4}$$

$$= \frac{P(F(D,t,j)|D)P(D)}{P(F(D,t,j)|D)P(D) + P(F(D,t,j)|\overline{D})P(\overline{D})}$$

where $P(F(D,t,j)|D)$ is the value of the PDF describing D calculated in the point $F(D,t,j)$, $P(F(D,t,j)|\overline{D})$ is the value of the PDF describing $\overline{D}$, $P(D)$ is the area of the distribution describing D and $P(\overline{D})$ is the area of the distribution for $\overline{D}$.

In order to estimate the parameters describing the PDF of D and $\overline{D}$ the Expectation Maximization (EM) algorithm for the Gaussian Mixture Model (Redner and Walker, 1984) is exploited. Assuming to model the empirical distribution of domain frequencies using a Gaussian mixture of two components, the estimated parameters can be used to evaluate domain relevance by equation (4).

3.3 The EM Algorithm for the GM model

In this section some details about the algorithm for parameter estimation are reported.

It is well known that a *Gaussian mixture* (GM) allows to represent every smooth PDF as a linear combination of normal distributions of the type in formula 5

$$p(x|\theta) = \sum_{j=1}^{m} a_j G(x, \mu_j, \sigma_j) \tag{5}$$

with

$$a_j \geq 0 \ \text{ and } \ \sum_{j=1}^{m} a_j = 1 \tag{6}$$

and

$$G(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{7}$$

and $\theta = \langle a_1, \mu_1, \sigma_1, \ldots, a_m, \mu_m, \sigma_m \rangle$ is a parameter list describing the gaussian mixture. The number of components required by the Gaussian Mixture algorithm for domain relevance estimation is $m = 2$.

Each component j is univocally determined by its weight a_j, its mean μ_j and its variance σ_j. Weights represent also the areas of each component, i.e. its total probability.

The Gaussian Mixture algorithm for domain relevance estimation exploits a Gaussian Mixture to approximate the empirical PDF of domain frequency scores. The goal of the Gaussian Mixture algorithm is to find the GM that maximize the likelihood on the empirical data, where the likelihood function is evaluated by formula (8).

$$L(\mathcal{T}, D, \theta) = \prod_{t \in \mathcal{T}} p(F(D,t)|\theta) \tag{8}$$

More formally, the EM algorithm for GM models explores the space of parameters in order to find the set of parameters θ such that the maximum likelihood criterion (see formula 9) is satisfied.

$$\theta_D = \underset{\theta'}{\operatorname{argmax}}\ L(\mathcal{T}, D, \theta') \tag{9}$$

This condition ensures that the obtained model fits the original data as much as possible. Estimation of parameters is the only information required in order to evaluate domain relevance for texts using the Gaussian Mixture algorithm. The Expectation Maximization Algorithm for Gaussian Mixture Models (Redner and Walker, 1984) allows to efficiently perform this operation.

The strategy followed by the EM algorithm is to start from a random set of parameters θ_0, that

has a certain initial likelihood value L_0, and then iteratively change them in order to augment likelihood at each step. To this aim the EM algorithm exploits a growth transformation of the likelihood function $\Phi(\theta) = \theta'$ such that $L(\mathcal{T}, D, \theta) \leqslant L(\mathcal{T}, D, \theta')$. Applying iteratively this transformation starting from θ_0 a sequence of parameters is produced, until the likelihood function achieve a stable value (i.e. $L_{i+1} - L_i \leqslant \epsilon$). In our settings the transformation function Φ is defined by the following set of equations, in which all the parameters have to be solved together.

$$\begin{aligned} \Phi(\theta) &= \Phi(\langle a_1, \mu_1, \sigma_1, a_2, \mu_2, \sigma_2 \rangle) \\ &= \langle a'_1, \mu'_1, \sigma'_1, a'_2, \mu'_2, \sigma'_2 \rangle \end{aligned} \quad (10)$$

$$a'_j = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \frac{a_j G(F(D,t_k), \mu_j, \sigma_j)}{p(F(D,t_k), \theta)} \quad (11)$$

$$\mu'_j = \frac{\sum_{k=1}^{|\mathcal{T}|} F(D,t_k) \cdot \frac{a_j G(F(D,t_k), \mu_j, \sigma_j)}{p(F(D,t_k), \theta)}}{\sum_{k=1}^{|\mathcal{T}|} \frac{a_j G(F(D,t_k), \mu_j, \sigma_j)}{p(F(D,t_k), \theta)}} \quad (12)$$

$$\sigma'_j = \frac{\sum_{k=1}^{|\mathcal{T}|} (F(D,t_k) - \mu'_j)^2 \cdot \frac{a_i G(F(D,t_k), \mu_i, \sigma_i)}{p(F(D,t_k), \theta)}}{\sum_{k=1}^{|\mathcal{T}|} \frac{a_j G(F(D,t_k), \mu_j, \sigma_j)}{p(F(D,t_k), \theta)}} \quad (13)$$

As said before, in order to estimate distribution parameters the British National Corpus (BNC-Consortium, 2000) was used. Domain frequency scores have been evaluated on the central position of each text (using equation 1, with c = 50).

In conclusion, the EM algorithm was used to estimate parameters to describe distributions for relevant and non-relevant texts. This learning method is totally unsupervised. Estimated parameters has been used to estimate relevance values by formula (4).

4 Domain Driven Disambiguation

DRE originates to improve the performance of Domain Driven Disambiguation (DDD). In this section, a brief overview of DDD is given. DDD is a WSD methodology that only makes use of domain information. Originally developed to test the role of domain information for WSD, the system is capable to achieve a good precision disambiguation. Its results are affected by a low recall, motivated by the fact that domain information is sufficient to disambiguate only "domain words". The disambiguation process is done comparing the domain of the context and the domains of each sense of the lemma to disambiguate. The selected sense is the one whose domain is relevant for the context[5].

In order to represent domain information we introduced the notion of Domain Vectors (DV), that are data structures that collect domain information. These vectors are defined in a multidimensional space, in which each domain represents a dimension of the space. We distinguish between two kinds of DVs: (i) *synset vectors*, which represent the relevance of a synset with respect to each considered domain and (ii) *text vectors*, which represent the relevance of a portion of text with respect to each domain in the considered set.

More formally let $\mathcal{D} = \{D_1, D_2, ..., D_d\}$ be the set of domains, the domain vector $\vec{s}$ for a synset s is defined as $\langle R(D_1, s), R(D_2, s), \ldots, R(D_d, s) \rangle$ where $R(D_i, s)$ is evaluated using equation (2). In analogy the domain vector $\vec{t_j}$ for a text t in a given position j is defined as $\langle R(D_1, t, j), R(D_2, t, j), \ldots, R(D_d, t, j) \rangle$ where $R(D_i, t, j)$ is evaluated using equation (4).

The DDD methodology is performed basically in three steps:

1. Compute $\vec{t}$ for the context t of the word w to be disambiguated
2. Compute $\hat{s} = argmax_{s \in Senses(w)} score(s, w, t)$ where
$$score(s, w, t) = \frac{P(s|w) \cdot sim(\vec{s}, \vec{t})}{\sum_{s \in Senses(w)} P(s|w) \cdot sim(\vec{s}, \vec{t})}$$
3. if $score(\hat{s}, w, t) \geqslant k$ (where $k \in [0, 1]$ is a confidence threshold) select sense $\hat{s}$, else do not provide any answer

The similarity metric used is the cosine vector similarity, which takes into account only the direction of the vector (i.e. the information regarding the domain).

$P(s|w)$ describes the prior probability of sense s for word w, and depends on the distribution of the sense annotations in the corpus. It is estimated by statistics from a sense tagged corpus (we used SemCor)[6] or considering the sense order in

[5] Recent works in WSD demonstrate that an automatic estimation of domain relevance for texts can be profitable used to disambiguate words in their contexts. For example, (Escudero et al., 2001) used domain relevance extraction techniques to extract features for a supervised WSD algorithm presented at the Senseval-2 competion, improving the system accuracy of about 4 points for nouns, 1 point for verbs and 2 points for adjectives, confirming the original intuition that domain information is very useful to disambiguate "domain words", i.e. words which are strongly related to the domain of the text.

[6] Admittedly, this may be regarded as a supervised component of the generally unsupervised system. Yet, we considered this component as legitimate within an unsupervised frame-

WORDNET, which roughly corresponds to sense frequency order, when no example of the word to disambiguate are contained in SemCor. In the former case the estimation of $P(s|w)$ is based on smoothed statistics from the corpus ($P(s|w) = \frac{occ(s,w)+\lambda}{occ(w)+|senses(w)|\cdot\lambda}$, where λ is a smoothing factor empirically determined). In the latter case $P(s|w)$ can be estimated in an unsupervised way considering the order of senses in WORDNET ($P(s|w) = \frac{2(|senses(w)|-sensenumber(s,w)+1)}{|senses(w)|(|senses(w)|+1)}$ where $sensenumber(s,w)$ returns the position of sense s of word w in the sense list for w provided by WORDNET.

5 Evaluation in a WSD task

We used the WSD framework to perform an evaluation of the DRE technique by itself.

As explained in Section 1 Domain Relevance Estimation is not a common Text Categorization task. In the standard framework of TC, categories are learned form examples, that are used also for test. In our case information in WORDNET DOMAINS is used to discriminate, and a test set, i.e. a corpus of texts categorized using the domain of WORDNET DOMAINS, is not available. To evaluate the accuracy of the domain relevance estimation technique described above is thus necessary to perform an indirect evaluation.

We evaluated the DDD algorithm described in Section 4 using the dataset of the Senseval-2 all-words task (Senseval-2, 2001; Preiss and Yarowsky, 2002). In order to estimate domain vectors for the contexts of the words to disambiguate we used the DRE methodology described in Section 3. Varying the confidence threshold k, as described in Section 4, it is possible to change the tradeoff between precision and recall. The obtained precision-recall curve of the system is reported in Figure 2.

In addition we evaluated separately the performance on nouns and verbs, suspecting that nouns are more "domain oriented" than verbs. The effectiveness of DDD to disambiguate domain words is confirmed by results reported in Figure 3, in which the precision recall curve is reported separately for both nouns and verbs. The performances obtained for nouns are sensibly higher than the one obtained for verbs, confirming the claim that domain information is crucial to disambiguate domain words.

In Figure 2 we also compare the results obtained by the DDD system that make use of the DRE technique described in Section 3 with the results obtained by the DDD system presented at the Senseval-2 competition described in (Magnini et al., 2002), that is based on the same DDD methodology and exploit a DRE technique that consists basically on the simply domain frequency scores described in subsection 3.1 (we refer to this system using the expression *old-DDD*, in contrast to the expression *new-DDD* that refers to the implementation described in this paper).

work since it relies on a general resource (SemCor) that does not correspond to the test data (Senseval all-words task).

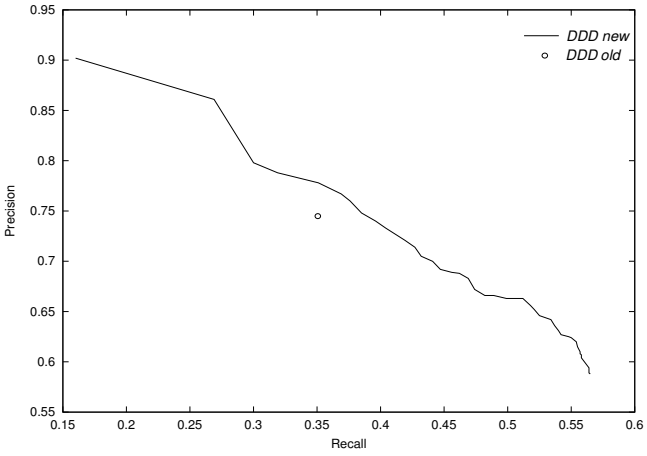

Figure 2: Performances of the system for all POS

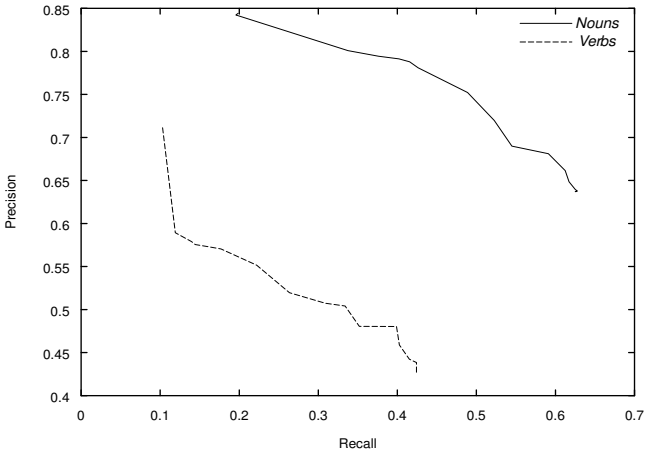

Figure 3: Performances of the system for Nouns and Verbs

Old-DDD obtained 75% precision and 35% recall on the official evaluation at the Senseval-2 English all words task. At 35% of recall the new-DDD achieves a precision of 79%, improving precision by 4 points with respect to old-DDD. At 75% precision the recall of new-DDD is 40%. In both cases the new domain relevance estimation technique improves the performance of the DDD methodology, demonstrating the usefulness of the DRE technique proposed in this paper.

6 Conclusions and Future Works

Domain Relevance Estimation, an unsupervised TC technique, has been proposed and evaluated inside the Domain Driven Disambiguation framework, showing a significant improvement on the overall system performances. This technique also allows a clear probabilistic interpretation providing an operative definition of the concept of domain relevance. During the learning phase annotated resources are not required, allowing a low cost implementation. The portability of the technique to other languages is allowed by the usage of synset-aligned wordnets, being domain annotation language independent.

As far as the evaluation of DRE is concerned, for the moment we have tested its usefulness in the context of a WSD task, but we are going deeper, considering a pure TC framework.

Acknowledgements

We would like to thank Ido Dagan and Marcello Federico for many useful discussions and suggestions.

References

BNC-Consortium. 2000. British national corpus, http://www.hcu.ox.ac.uk/BNC/.

J. P. Comaroni, J. Beall, W. E. Matthews, and G. R. New, editors. 1989. *Dewey Decimal Classification and Relative Index*. Forest Press, Albany, New York, 20^{th} edition.

G. Escudero, L. Màrquez, and G. Rigau. 2001. Using lazy boosting for word sense disambiguation. In *Proc. of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation System*, pages 71–74, Toulose, France, July.

C. Fellbaum. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

Y. Ko and J. Seo. 2000. Automatic text categorization by unsupervised learning. In *Proceedings of COLING-00, the 18^{th} International Conference on Computational Linguistics*, Saarbrücken, Germany.

B. Magnini and G. Cavaglià. 2000. Integrating subject field codes into WordNet. In *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, Athens, Greece, June.

B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4):359–373.

M. Palmer, C. Fellbaum, S. Cotton, L. Delfs, and H.T. Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, Toulouse, France, July.

J. Preiss and D. Yarowsky, editors. 2002. *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, Toulouse, France.

R. Redner and H. Walker. 1984. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, April.

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

Senseval-2. 2001. http://www.senseval.org.