Unsupervised and Supervised Exploitation of Semantic Domains in Lexical Disambiguation [1]

Alfio Gliozzo [a] Carlo Strapparava [a,*] Ido Dagan [b]

[a] *ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica, I-38050, Trento, Italy*

[b] *Computer Science Department, Bar Ilan University, Ramat Gan, Israel*

Abstract

Domains are common areas of human discussion, such as economics, politics, law, science etc., which are at the basis of lexical coherence. This paper explores the dual role of domains in word sense disambiguation (WSD). On one hand, domain information provides generalized features at the paradigmatic level that are useful to discriminate among word senses. On the other hand, domain distinctions constitute a useful level of coarse grained sense distinctions, which lends itself to more accurate disambiguation with lower amounts of knowledge.

In this paper we extend and ground the modeling of domains and the exploitation of WordNet Domains, an extension of WordNet in which each synset is labeled with domain information. We propose a novel unsupervised probabilistic method for the critical step of estimating domain relevance for contexts, and suggest utilizing it within unsupervised Domain Driven Disambiguation (DDD) for word senses, as well as within a traditional supervised approach.

The paper presents empirical assessments of the potential utilization of domains in WSD at a wide range of comparative settings, supervised and unsupervised. Following the dual role of domains we report experiments that evaluate both the extent to which domain information provides effective features for WSD, as well as the accuracy obtained by WSD at domain-level sense granularity. Furthermore, we demonstrate the potential for either avoiding or minimizing manual annotation thanks to the generalized level of information provided by domains.

Key words:
Word Sense Disambiguation, Semantic Domains, WordNet, Unsupervised Learning, Lexical Resources.

[*] Corresponding author.
Email addresses: gliozzo@itc.it (Alfio Gliozzo), strappa@itc.it (Carlo Strapparava), dagan@cs.biu.ac.il (Ido Dagan).

[1] This work was developed under the collaboration ITC-irst/University of Haifa.

1 Introduction and Motivations

Domains are common areas of human discussion, such as economics, politics, law, science, etc. (see Table 1), which demonstrate lexical coherence. A substantial portion of the language terminology may be characterized as *domain words* whose meaning refers to concepts belonging to specific domains, and which often occur in texts that discuss the corresponding domain.

Domains have been used with a dual role in linguistic description. One role is characterizing word senses, typically as the semantic field of a word sense in a dictionary or lexicon (e.g. *crane* has senses in the domains of ZOOLOGY and CONSTRUCTION). The WORDNET DOMAINS lexical resource is an extension of WORDNET which provides such domain labels for all synsets (Magnini and Cavaglià, 2000). A second role is to characterize texts, typically as a generic level of text categorization (e.g. for classifying news and articles) (Sebastiani, 2002).

From the perspective of word sense disambiguation domains may be considered from two points of view. First, a major portion of the information required for sense disambiguation corresponds to paradigmatic domain information. Many of the features that contribute to disambiguation identify the domains that characterize a particular sense or subset of senses. For example, economics terms provide characteristic features for the financial senses of words like *bank* and *interest*, while legal terms characterize the judicial sense of *sentence* and *court*. Common supervised WSD methods capture such domain-related distinctions separately for each sense of each word, and may require relatively many training examples in order to obtain sufficiently many features of this kind for each sense (Yarowsky and Florian, 2002). However, domains represent an independent linguistic notion of discourse, which does not depend on a specific word sense. Therefore, it is beneficial to model a relatively small number of domains directly, as a generalized notion, and then use the same generalized information for many instances of the WSD task. A major goal of this paper is to study the extent to which domain information can contribute along this vein to WSD.

Second, domains may provide a useful coarse-grained level of sense distinctions. Many applications do not benefit from fine grained sense distinctions (such as WORDNET synsets), which are often impossible to detect by WSD within practical applications (i.e. some verbs in WORDNET have more than 40 sense distinctions). However, applications such as information retrieval (Gonzalo et al., 1998) and user modeling for news web sites (Magnini and Strapparava, 2001) can benefit from sense distinctions at the domain level, which are substantially easier to establish in practical WSD.

The work by Magnini et al. (2002) has presented initial results in utilizing WordNet Domains information for WSD (at the WordNet synset sense granularity level). In this paper we substantially extend and ground domain modeling and the utilization of WordNet Domains in several ways. At the algorithmic level, we present a novel unsupervised method for estimating domain relevance for word contexts, which is grounded in a probabilistic framework utilizing Gaussian Mixtures and EM estimation. The unsupervised estimation framework, which is very attractive for WSD, is made possible thanks to the dual nature of domains being both sense and text descriptors. This enables us to use only the available lexical resource of WordNet Domains without requiring annotated examples. The focus of this paper is not about the absolute performance of a particular new WSD system, but rather to investigate and assess the potential utilization of domains in WSD at a wide range of comparative settings, both supervised and unsupervised. In particular we report experiments that evaluate:

- the extent to which domain information provides effective features for WSD;
- the accuracy that can be obtained by WSD at domain-level sense granularity;
- the potential for avoiding or minimizing manual annotation thanks to the generalized information provided by domains.

The paper is structured as follows. Section 2 describes the notion of semantic domains and some prior work. Section 3 presents the lexical resource WordNet Domains. Section 4 lays the grounds for the computational modeling of domains using WordNet Domains. In particular the notion of domain relevance for both the textual and lexical levels is introduced. Section 5 presents the computational methods by which semantic domains can be exploited within WSD. Section 6 presents the experiments and evaluation, and Section 7 suggests conclusive remarks.

2 Background: Semantic Domains and Sense Discrimination

This section introduces the notion of semantic domains from linguistic and computational perspectives, suggesting that semantic domains provide a useful component for modeling lexical ambiguity.

The problem of describing word senses is addressed by lexicographers using either definitions of concepts (e.g. a dictionary) or reporting a set of words that are related to the concept to be described (e.g. a thesaurus or the WordNet structure). We refer to the latter type as "relational definitions".
Relations between words can be classified into two main groups, namely *paradigmatic* and *syntagmatic* relations (de Saussure, 1922). Two words are syn-

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.

tagmatically related when they frequently appear in the same syntagm, e.g. when one of them frequently follows the other. Two words are paradigmatically related when their meanings are very closely related, like synonyms and hyponyms.

Lexical polysemy corresponds to the fact that different senses have different relational definitions, that is, different lists of syntagmatically and paradigmatically related terms. Thus, different senses of a term are described by their typical collocation relations (syntagmatic polysemy) or by their typical domains of usage (paradigmatic polysemy or domain polysemy). For example, the lemma *bank*, as a noun, has 10 different senses (see Table 2): most of them can be differentiated by only considering domain distinction. It is important to notice here that domain information for a word sense indicates the typical domain of the texts in which the sense occurs. Hence, domain information constitutes both a property (class) of the word itself, as well as a linguistic feature that characterizes the text (unlike a word sense, which is only a property of the word).

Syntagmatic and paradigmatic relations, which have been modeled symbolically in the lexicographic tradition, are often modeled by statistical information in computational frameworks. Syntagmatic relations have been estimated in various ways; for example using mutual information between words, building language models or studying collocations. Paradigmatic relations seem harder to model statistically, due to data sparseness and because this type of information often has a somewhat fuzzy nature. As will be discussed below, domain information corresponds to a paradigmatic relationship and can provide an effective mean for its modeling, at the text and term levels.

2.1 Semantic domains and their use

From a practical point of view, semantic domains are considered as a list of related words describing a particular subject or area of interest. Domain oriented words are typically highly correlated within texts, i.e. they tend to co-occur inside the same types of texts, supporting lexical coherence. Many dictionaries, as for example LDOCE (Procter, 1978), indicate domain specific usages by attaching Subject Field Codes to word senses. Although this type of information is useful for sense discrimination, dictionaries often specify subject codes only for a small portion of the lexicon, leaving most of the senses unlabeled with respect to their semantic field.

A prominent linguistic work about semantic domains is the Semantic Fields Theory (Trier, 1934), proposed by Jost Trier in the 1930's. A Semantic Field consists of a structured set of very closely related concepts, lexicalized by a set of domain specific terms. The meaning of these terms are determined and delimitated only by the terms inside the same semantic field.

In the NLP literature the exploitation of Semantic Fields has been shown fruitful for sense disambiguation (e.g. the pioneering works of (Guthrie et al., 1991; Yarowsky, 1992)). Guthrie et al. (1991) exploited the *subject-codes* supplementary fields of LDOCE. In addition to using the Lesk-based method of counting overlaps between definitions and contexts, they imposed a correspondence of subject codes in an iterative process. Yarowsky (1992) bases WSD on the 1,042 categories of the Roget's Thesaurus. The idea underlying the algorithm is that different word senses tend to belong to different conceptual classes, and such classes tend to appear in recognizably different contexts. From a technical point of view, the correct subject category is estimated maximizing the sum of a Bayesian term ($log\frac{Pr(word|RCat)}{Pr(word)}$ - i.e. the probability of a word appearing in a context of a Roget category divided by its overall probability in the corpus) over all possible subject categories for the ambiguous word in its context (± 50 words). Thus, identifying the conceptual class of a context provides a crucial clue for discriminating word senses tha belong to that class. The results were promising, the system correctly disambiguated 92% of the instances of 12 polysemous words on the Grolier's Encyclopedia. Stevenson and Wilks (1999) use the LDOCE subject codes by adapting the Yarowsky algorithm. Experiments were performed on a subset of the British National Corpus (BNC) on the words appearing at least 10 times in the training context of a particular word. In addition, while Yarowsky (1992) assumed a uniform prior probability for each Roget category, the probability of each subject category was estimated as the proportion of senses in LDOCE to which a given category was assigned.

More recently (Escudero et al., 2000) used domain features extracted from WordNet Domains in a supervised classification setting tested on the Senseval-2 tasks. Prior probabilities for each domain were computed considering the frequency of a domain. The introduction of such domain features systematically improved the system performance, especially for nouns (over three percentage points of improvement). While Escudero et al. (2000) integrated domains within a wider set of features; Magnini et al. (2001) presented a system completely based on domain information at Senseval-2. The underlying hypothesis of the approach is that information provided by domain labels offers a natural way to establish associations among word senses in a certain text fragment, which can be profitably used during the disambiguation process.

A common problem of many previous attempts to utilize semantic domains in WSD is that very frequent words have, in general, many senses belonging to different domains. Thus, all methods based on simple frequency counting often turn out to be inadequate: irrelevant senses of ambiguous words contribute to increase the final score of irrelevant domains, introducing noise. Moreover, the level of noise is different for different domains because of their different sizes and potential differences in the ambiguity level of their vocabularies. In order to discriminate between noise and relevant information it is possible to use a supervised framework, exploiting labeled training data. But unfortunately domain labeled text corpora are not easily available. To overcome this problem, in this paper (see Section 4.2.3) we propose a Gaussian Mixture approach, that constitutes an unsupervised way to distinguish in texts relevant domain information from noise.

3 WordNet Domains

WordNet Domains[2] is an extension of WordNet (Fellbaum, 1998), in which each synset is annotated with one or more domain labels. About 200 domain labels were selected from a number of dictionaries and then structured in a taxonomy according to their position in the (much larger) Dewey Decimal Classification system (DDC), which is commonly used for classifying books.

[2] Freely available for research from `http://wndomains.itc.it`

Sense	Synset and Gloss	Domains	Semcor
#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", as a noun.

DDC was chosen because it ensures good coverage, is easily available and is commonly used to classify "text material" by librarians. Finally, it is officially documented and the interpretation of each domain is detailed in the reference manual (Comaroni et al., 1989).

Domain labeling of synsets is complementary to the information already in WordNet. First, a domain may include synsets of different syntactic categories: for instance Medicine groups together senses of 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 derived from different "unique beginners" or from different "lexicographer files" [3]). For example, Sport contains senses such as `athlete#1`, derived from `life_form#1`, `game_equipment#1` from `physical_object#1`, `sport#1` from `act#2`, and `playing_field#1` from `location#1`.

[3] The noun hierarchy is a tree forest, with several roots (*unique beginners*). The *lexicographer files* are the source files from which WordNet is "compiled". Each lexicographer file is usually related to a particular topic.

The annotation methodology (Magnini and Cavaglià, 2000) was primarily manual and was based on lexico-semantic criteria that take advantage of existing conceptual relations in WordNet. First, a small number of high level synsets were manually annotated with their pertinent domain. Then, an automatic procedure exploited some of the WordNet relations (i.e. hyponymy, troponymy, meronymy, antonymy and pertain-to) to extend the manual assignments to all the reachable synsets. For example, this procedure labeled the synset {`beak, bill, neb, nib`} with the code Zoology through inheritance from the synset {`bird`}, following a "part-of" relation. However, there are cases in which the inheritance procedure was blocked, by inserting "exceptions", to prevent incorrect propagation. For instance, `barber_chair#1`, being a "part-of" `barbershop#1`, which in turn is annotated with Commerce, would wrongly inherit the same domain. An evaluation of the annotation has been carried on by means of a text classification task (see (Magnini and Cavaglià, 2000)). The entire process had cost approximately 2 person-years.

Domains may be used to group together senses of a particular word that have the same domain labels. Such grouping reduces the level of word ambiguity when disambiguating to a domain, as demonstrated in Table 2. The noun "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` belong to both Geography and Geology. Grouping related senses in order to achieve more "practical" coarse-grained senses is an emerging topic in WSD (see, for instance (Palmer et al., 2001)). Domain granularity was used in our experiments to evaluate disambiguation performance at a coarse-grained level.

In the remainder of this paper we employ a concrete vector-based representation of domain information. Domain vectors are defined in a multidimensional space, where each domain corresponds to one dimension. We chose to use a subset of the domain labels (Table 1) in WordNet Domains (see Section 3). For example, Sport is used instead of Volley or Basketball, which are subsumed by Sport. This subset was selected empirically to allow a sensible level of abstraction without losing much relevant information, overcoming data sparseness for less frequent domains. Principled selection (or construction) of the most optimal set of domains for WSD is beyond the scope of this paper, and is left as an open issue for future research.

Finally, some WordNet synsets do not belong to a specific domain but rather correspond to general language and may appear in any context. Such senses are tagged in WordNet Domains with a Factotum label, which may be considered as a "placeholder" for all other domains. Accordingly, Factotum is not one of the dimensions in our domain vectors, but is rather reflected as a property of those vectors which have a relatively uniform distribution across all domains.

4 Computational Modeling of Domains

As already highlighted, domains have a dual role in describing both the lexicon and the texts. This section introduces the notion of *domain vectors* for concepts (senses), words, and texts, along with *domain relevance* estimation which provides the values of vector entries.

4.1 Domain vectors

Domain vectors (DVs) are defined in a d dimensional space, where d is the cardinality of the domain set. The value of each component (dimension) is the relevance value of the corresponding domain with respect to the object described by the vector. DVs are defined for three types of objects: (i) *concept vectors*, representing domain relevance values for concepts (i.e. WordNet synsets, corresponding to word senses); (ii) *word vectors*, representing domain relevance values for words; and (iii) *text vectors*, represent domain relevance values for a window of text around the disambiguated word occurrence.

Typically, DVs related to generic senses (Factotum synsets) have a flat distribution, while DVs for domain specific senses are strongly oriented along one dimension. We hypothesize that to ensure coherence many of the words in a given text have to be domain oriented, supporting their reciprocal disambiguation. Otherwise stated, words taken out of context show domain polysemy, but when they are used inside real texts domain polysemy is largely reduced, and only one or few domains emerge in each text vector[4]. This observation fits with the general lexical coherence assumption, viewed in our setting as domain coherence, and is exploited by the Domain Driven Disambiguation (DDD) method (Section 5.1).

As common for vector representations, DVs enable us to compute domain similarity between objects of either the same or different types (between texts, between a concept and a text, and between concepts) using similarity metrics over the same vectorial space. This property suggests the potential of utilizing domain similarity between various types of objects for different NLP tasks.

[4] Intuitively, texts may exhibit somewhat stronger or weaker orientation towards specific domains, but it seems less sensible to have a text that is not related to at least one domain. In other words, it is difficult to find a "generic" (Factotum) text. This intuition is largely supported by our data, where every text in the corpus exhibits a small number of relevant domains, demonstrating the property of domain coherence for texts. In (Magnini et al., 2002) a "one domain per discourse hypothesis" was proposed and verified on SemCor, the portion of the Brown corpus semantically annotated with WordNet senses.

For example, measuring the similarity between the DV of a word context and the DVs of its alternative senses is useful for WSD, as demonstrated in this paper. Measuring the similarity between DVs of different texts may be useful for domain-oriented text clustering, and so on.

4.2 Domain Relevance

The domain relevance function R of a domain D with respect to a linguistic object o - text, word or concept (a synset, corresponding to a sense) - quantifies (weighs) the degree of association between D and o. R obtains positive real values, where a higher value indicates a higher degree of relevance. In most of our settings the relevance value ranges in the interval $[0,1]$, but this is not a necessary requirement.

We next present the methods used to compute domain relevance for concepts (Section 4.2.1), words (Section 4.2.2) and texts (Section 4.2.3). While the computation of domain relevance for concepts and words is relatively straightforward, we propose an unsupervised algorithm to estimate domain relevance for texts in a normalized probabilistic manner. These estimates are then used to refine domain relevance for concepts in supervised WSD settings (Section 4.2.4).

4.2.1 Domain relevance for concepts

Following the WordNet approach, we assume that each word sense corresponds to a particular *concept*, represented as a WordNet synset. Intuitively, a domain D is relevant for a concept c if D is relevant for the texts in which c usually occurs. As an approximation the information in WordNet Domains can be used to estimate such a function. Let $\mathcal{D} = \{D_1, D_2, ..., D_d\}$ be the set of domains, $C = \{c_1, c_2, ..., c_k\}$ be the set of concepts (synsets) and $R : \mathcal{D} \times C \Rightarrow [0,1]$ be the domain relevance function for concepts.

The domain assignment to synsets from WordNet Domains is represented by the function $Dom : C \Rightarrow P(D)$ [5], which returns the set of domains associated with each synset c. Formula 1 defines the domain relevance estimation function (recall that d is the domain set cardinality):

$$R(D,c) = \begin{cases} 1/|Dom(c)| & : \texttt{if } D \in Dom(c) \\ 1/d & : \texttt{if } Dom(c) = \{\textsc{Factotum}\} \\ 0 & : \texttt{otherwise} \end{cases} \quad (1)$$

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

$R(D,c)$ can be perceived as an estimated prior for the probability of the domain given the concept, according to the WordNet Domains annotation. Under these settings Factotum (generic) concepts have uniform and low relevance values for each domain while domain oriented concepts have high relevance values for a particular domain. For example, given Tables 1 and 2, $R(\textsc{Economy}, \texttt{bank\#5}) = 1/42$, $R(\textsc{Economy}, \texttt{bank\#1}) = 1$, and $R(\textsc{Economy}, \texttt{bank\#8}) = 1/2$. Notice that this estimation depends only on the available resource of WordNet Domains, and does not require supervised labeled data. In Section 4.2.4 we present a refined method for estimating domain relevance for concepts that utilizes annotated WSD training examples in a supervised setting.

4.2.2 Domain relevance for words

Domain relevance for a word is derived directly from the domain relevance values of its senses. Intuitively, a domain D is relevant for a word w if D is relevant for one or more senses c of w. Let $V = \{w_1, w_2, ...w_{|V|}\}$ be the vocabulary, let $senses(w) = \{c | c \in C, c \text{ is a sense of } w\}$ (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 the average relevance value of its senses:

$$R(D, w) = \frac{1}{|senses(w)|} \sum_{c \in senses(w)} R(D, c) \tag{2}$$

Notice that domain relevance for a monosemic word is equal to the relevance value of the corresponding concept. A word with several senses will be relevant for each of the domains of its senses, but with a lower value. Thus monosemic words are more domain oriented than polysemic ones and provide a greater amount of domain information. This phenomenon often converges with the common property of less frequent words being more informative, as they typically have fewer senses.

This framework provides also a formal definition of *domain polysemy* for a word w, defined as the number of different domains belonging to w's senses: $P(w) = |\bigcup_{c \in senses(w)} Dom(c)|$. We propose using such coarse grained sense distinction for WSD, enabling to obtain higher accuracy for this easier task (Section 6.3). Initial work on using domain grained distinctions for WSD over parallel corpora is reported in (Magnini and Strapparava, 2000).

4.2.3 Domain relevance for texts

It is now possible to define the notion of domain relevance for texts. Intuitively, a domain D is relevant for a text t if D is relevant for the words in t. Let

$T = \{t_1, t_2, ..., t_m\}$ be a set of texts, with the domain relevance function $R : \mathcal{D} \times T \Rightarrow [0, 1]$. For example, the domain relevance of ECONOMY for the text *"He cashed a check at the bank"* is expected to be very close to 1, while the domain relevance for an unrelated domain like SPORT is 0.

Domain relevance estimation for a text relies on lexical domain coherence, having a substantial portion of the text words associated with the same domain. An initial step to capture this property is accumulating ("counting") domain relevance for the text words over all domains. In the context of WSD we consider a weighted window of text around the word to be disambiguated. Such local estimation of domain relevance is important in order to take into account possible domain shifts along the text (Magnini et al., 2002).

Let t be a text window containing c words on each side of the word to be disambiguated, where the word indices run from $-c$ to c, i.e. $t = w_{-c}, \ldots, w_c$. For each domain D a "frequency" score in t is computed as follows:

$$F(D,t) = \sum_{i=-c}^{c} R(D, w_i) G(i, 0, (\frac{c}{2})^2) \tag{3}$$

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.

Unfortunately the raw frequency score is in practice not a good domain relevance measure, mainly due to the noise introduced by lexical polysemy. In particular, frequent words typically have many senses belonging to different domains, and it is not possible to assume knowing their actual sense in advance within a WSD framework. Consequently, irrelevant senses of ambiguous words contribute to augment the final frequency score of irrelevant domains, introducing significant noise. Moreover, the level of noise is substantially different for different domains, due to differences in their sizes and in the ambiguity level of their vocabularies. For example (see Table 1) the number of PSYCHOLOGY synsets is more than 30 times greater than the number of VETERINARY synsets, yielding a much higher level of noise. As a result, relatively high frequency scores for PSYCHOLOGY may still correspond to irrelevant texts, while relatively low scores for VETERINARY will suffice to indicate domain relevancy.

In order to obtain an effective domain relevancy measure (to be utilized for WSD in the next section) it is necessary to discriminate between noise and relevant information. One option is to use a supervised framework in which significance levels for frequency scores of each domain can be estimated from labeled training data. However, this would require a substantial quantity of domain labeled texts (for each domain), which are typically not available. As an alternative we propose a completely unsupervised solution based on Gaussian Mixtures (GM), which differentiates relevant domain information from noise based on statistical estimation from raw unlabeled texts.

The underlying assumptions of the Gaussian Mixture approach are that frequency scores for a certain domain are obtained from an underlying (unknown) mixture of relevant and non-relevant texts, and that the scores for relevant texts are significantly higher than scores of non-relevant ones. The frequency scores are thus 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[6] (BNC-Consortium, 2000) using formula 3. The "empirical" PDF, describing the distribution of frequency scores in the training corpus, is represented by the continuous line.

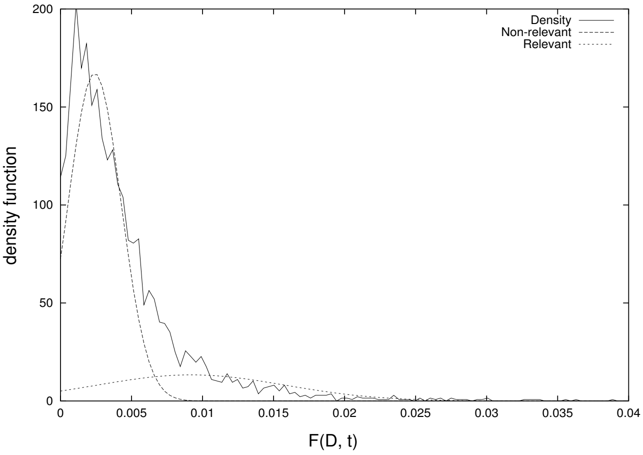

Fig. 1. Gaussian mixture for D = Sport

Examining the graph suggests 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 of (lower) frequency scores for the majority of non relevant texts ($\overline{D}$); the smaller distribution on the right is assumed to be the distribution of (higher) frequency scores for the minority of relevant texts (D).

$\overline{D}$ describes the noise within frequency estimation counts, produced by polysemous words and by occasional occurrences of Sport terms in non-relevant texts. The goal of the GM technique is to estimate the parameters of the two distributions in order to assign high domain relevance values only for truly relevant frequency scores. It is reasonable to assume that $\overline{D}$ 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. D, on the other hand, describes typical frequency values for relevant texts. This distribution is also assumed to be normal.

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

A probabilistic interpretation enables to evaluate the relevance value $R(D,t)$ for a domain D and a text window t by considering only the domain frequency $F(D,t)$. It is defined as the conditional probability $P(D|F(D,t))$, estimated as follows using Bayes theorem:

$$\begin{aligned} R(D,t) &= P(D|F(D,t)) \\ &= \frac{P(F(D,t)|D)P(D)}{P(F(D,t)|D)P(D) + P(F(D,t)|\overline{D})P(\overline{D})} \end{aligned} \tag{4}$$

where $P(F(D,t)|D)$ is the value of the PDF of D at the point $F(D,t)$, $P(F(D,t,j)|\overline{D})$ is the value of the PDF of $\overline{D}$ at the same point, $P(D)$ is the area of the D distribution and $P(\overline{D})$ is the area of the $\overline{D}$ distribution. The parameters of the distributions D and $\overline{D}$, used to evaluate equation 4, are estimated by the Expectation Maximization (EM) algorithm for the Gaussian Mixture Model (Redner and Walker, 1984). Details of this algorithm are provided in appendix A.

In summary, the Gaussian Mixture approach allows principled unsupervised estimation of a probabilistic domain relevance value. Probabilistic estimation yields a uniform scale of relevance values for all domains, regardless of their size and the inherent level of noise in their vocabularies.

4.2.4 Supervised acquisition of domain relevance for concepts

As explained earlier the domain vector for a concept contains all the domain relevance values for that concept. A domain D is considered relevant for a concept c if c typically occurs in texts t belonging to D, i.e. in those texts for which we expect a high value of $R(D,t)$. In Section 4.1 domain relevance for concepts was estimated from the domain assignments to concepts in WordNet Domains, assuming that these lexicographic assignments reflect domain relevance properly. In this section we consider a supervised WSD setting, in which concept labels are available for word occurrences in training texts. Such labeled data enable us to improve empirically domain relevancy estimation for concepts, by examining the actual domain relevancy values $R(D,t)$ for all texts in which the concept c occurs.

Let T_c be the set of all text windows in the training corpus that are centered around a word which is labeled by the concept c. The supervised estimate for $R(D,c)$ is defined by[7]:

[7] We do not care here for normalizing the $R(D,c)$ values by the frequency of c since domain relevance vectors for concepts are normalized anyhow within our disambiguation method, through the cosine vector similarity measure (Section 5.1).

$$R(D,c) = \sum_{t \in T_c} R(D,t) \qquad (5)$$

We note some relevant aspects of the above estimation. The direction of the domain vector indicates the major domain (or few domains) of the concept, i.e. the “typical” domain(s) of the texts in which the concept occurs. Thus, vectors of concepts that occur mostly in texts of the same domain will be clearly oriented towards that domain. Vectors of generic concepts, which occur in texts of arbitrary domains, will be relatively “flat” with a low value in each domain. Sufficient training data per concept is needed to obtain a proper vector orientation, in particular for generic (Factotum) concepts.

5 Exploiting semantic domains for WSD

WSD is the task of determining the meaning of a word in context. The domain modeling presented in Section 4 suggests several roles for domain information within WSD:

(1) domains are properties of texts, providing a natural description for the context of the word to disambiguate;
(2) domains are properties of word senses, providing effective features for sense models;
(3) domain information provides groupings of word senses, which may be used as coarse-grained target senses for WSD.

This section presents two alternative methods for exploiting our domain modeling in WSD, realizing the first two roles above. These methods are then evaluated in Section 6, on both fine-grained and coarse-grained sense distinctions.

5.1 Domain Driven Disambiguation

Domain Driven Disambiguation (DDD) (Magnini et al., 2002) is a generic WSD methodology that utilizes only domain information. First, domain relevance vectors for concepts, $DV(c)$, are computed in a pre-processing phase. Then, during disambiguation, the following three steps are performed for each occurrence of an ambiguous word w:

- Compute the domain relevance vector $DV(t)$ for the text window t around w

- Compute $\hat{c} = argmax_{c \in Senses(w)} score(c, w, t)$ where

$$score(c, w, t) = \frac{P(c|w) \cdot sim(DV(c), DV(t))}{\sum_{c \in Senses(w)} P(c|w) \cdot sim(DV(c), DV(t))}$$

  where $sim(DV(c), DV(t))$ is some vector similarity metric.
- if $score(\hat{c}, w, t) \geqslant k$ (where $k \in [0, 1]$ is a confidence threshold) select sense $\hat{c}$, else do not provide any answer

Our implementation of the DDD methodology utilizes the Gaussian Mixture algorithm (Section 4.2.3) to evaluate domain relevance for the text window t[8]. Following Gale et al. (1992), the size of the context used to estimate domain relevance (i.e. the parameter c in formula 3) has been fixed empirically at 50. The commonly used cosine measure was chosen as the vector similarity metric sim.

Domain relevance for concepts can be computed in either an unsupervised or supervised mode, yielding two versions of the WSD system. In the unsupervised version domain relevance for concepts is based on the information in WordNet Domains, as in Section 4.2.1. In the supervised version, the domain vectors were learned from sense labeled training data, as in Section 4.2.4. We refer to these versions as *Unsupervised DDD* and *Supervised DDD*, respectively. Generally, the unsupervised DDD was used in the “all-words” tasks (see Section 6), for which training data is not provided, while supervised DDD was used in the “lexical-sample” tasks, where training data for each test word are available.

Finally, $P(c|w)$ describes the prior probability of a sense c for the word w, which may depend on the distribution of senses in the target corpus. We estimated these probabilities based on frequency counts in the generally available SemCor corpus[9].

[8] The original implementation in (Magnini and Strapparava, 2000) utilized an adhoc (and less accurate) procedure which was based directly on domain frequency counts.

[9] Admittedly, this may be regarded as a supervised component within the generally unsupervised system version. However, we considered this component as legitimate within a typical unsupervised setting since it relies on a general resource (SemCor) that does not correspond to the test data and task, as in the all-words task. This setting is distinguished from having quite a few annotated training examples that are provided by construction for each test sense in a supervised setting, as in the Lexical Sample task.

5.2 Domains as features for supervised WSD

Many state of the art WSD systems are designed within a supervised machine learning paradigm, in which disambiguation is approached as a classification task. Each word is disambiguated by constructing an independent classifier, for which a specific learner is trained from annotated examples of the given word; this approach is also known as the *word expert* approach. Each example word occurrence is represented by a set of features extracted from the context of the target word. Feature extraction is a very delicate part of the WSD process: depending on the features selected to describe a word occurrence the quality of the learned sense model can vary substantially.

It is important to point out that a mixture of very different features are typically used. A widely accepted classification of the features typically used in WSD (Florian et al., 2002; Yarowsky and Florian, 2002) consists of the dichotomy between local and topical features. Following the terminology introduced in Section 2, local features largely consist of syntagmatic relations, while topical features can be used to capture paradigmatic relations.

Local features include words in local context, bigrams, trigrams and syntactic features such as POS and verb-object relations. Topical features are typically represented using a *bag of words* (BOW) approach.

The BOW approach presents several disadvantages in modeling paradigmatic properties. The most important one is the data sparseness in the vectors representing texts, whose number of dimensions is equal to the vocabulary size. Such sparse data requires a very large number of training examples to discover sufficiently many context word features for all senses in the lexicon. Unfortunately this is not the case for WSD, where training data is not available for most senses, and is limited to no more than 10-20 examples per word sense in just a few available annotated corpora.

A typical solution for such data sparseness is reducing feature space dimensionality. Using domains as paradigmatic features is conceived in this perspective. This is realized in our framework by evaluating domain relevance for the text window surrounding each ambiguous word occurrence, as presented in Section 4.2.3. Then, the most relevant domains, such as those whose relevance score exceeds a threshold, can be selected as (binary) features that describe the word context. Methods that consider a continuous weight for the "strength" of features within examples may utilize the actual value of domain relevance for the text as such weight. In both cases, our probabilistic relevance score provides a uniform and consistent scale for feature selection and weighting across all domains.

Our hypothesis, supported by the empirical results in the next section, is that domain information is largely equivalent to a BOW representation of the word context. Consequently, the BOW feature set can be substituted by domain features, as estimated for the text window[10]. Using this approach has several advantages. First, the information in WordNet Domains can be exploited in order to obtain refined domain relevance estimations; second, dimensionality reduction of the feature space allows to reduce the amount of labeled training needed to model paradigmatic relations, while the estimation of the GM model of domain relevance for texts utilizes the information in additional unlabeled texts; finally, the usage of a lexical resource to model the feature space, if correctly done, provides the classification algorithm with yet more information than is available from the training examples alone.

6 Experiments and Evaluation

This section reports an empirical evaluation of using domain information in WSD. In particular the following claims are assessed:

(1) domains provide informative paradigmatic features to model sense distinctions,
(2) domain level granularity for sense distinction enables to obtain high disambiguation accuracy,
(3) Domain Driven Disambiguation is a practical unsupervised methodology to exploit WSD in real world applications.

To support claim 1 domain labels were used as features in a supervised WSD setting, as described in Section 5.2. The supervised algorithm was tested on the Senseval-2 English Lexical Sample task using different feature sets. Learning curves and Precison-Coverage curves are reported.

To support claim 2 the DDD and the supervised systems have been tested on both the Senseval-2 English Lexical Sample and All Words tasks using domain granularity sense distinctions. The supervised system performance was compared to the unsupervised one.

To support claim 3 the DDD system has been tested on the Senseval-2 All Words task. These experiments exploited mainly the information contained in WordNet Domains as well as estimating prior sense probabilities from SemCor (as explained in Section 5.1), but without using annotated examples to learn supervised disambiguation models.

[10] In fact, we have found that using both domains and BOW features decreases WSD performance. This result may seem surprising, but it may be explained as an over-emphasis of incorrect classifications when using both types of features.

Section 6.1.1 describes the WSD task and Section 6.1.2 presents the WSD systems used in the experiments. The following three sections present the evaluation of the above three claims.

6.1 Evaluation Framework

6.1.1 WSD evaluation tasks

The following three corpora were used in our experiments:

[**ALL**] *Senseval-2*[11] *English all-words task*: the test data for the English all-words task consists of 5,000 words of running text from three Penn Treebank II Wall Street Journal articles. The total number of words that have to be disambiguated is 2,473. Sense tags are assigned using WordNet 1.7. Training examples are not provided in this collection.

[**LEX**] *Senseval-2 English lexical sample task*: the test data were collected, for the most part, from the BNC corpus (adjectives and nouns) and from the Wall Street Journal corpus (for verbs), for a total of more than 500,000 words. This gold standard consists of small texts each of them about 120 words long. In each text an instance to be disambiguated is present. The total instances to be disambiguated are 4,328. Sense tags are assigned using WordNet 1.7. In this collection labeled training examples are provided for each word that is included in the test set. There were 29 nouns, 29 highly polysemous verbs, and 15 adjectives, with between 70 and 455 instances per word (divided 2:1 between training data and test data).

6.1.2 The WSD systems used for evaluation

We used both *Unsupervised DDD* and *Supervised DDD* (Section 5.1) in the experiments. Typically unsupervised DDD has been used in the "all-words" tasks, for which training data is not available, while supervised DDD has been used in the "lexical-sample" tasks, where training data for each word to disambiguate are available.

To test the use of domain features in a supervised WSD framework we chose to implement a *decision list* (DL) algorithm (Resnik and Yarowsky, 1997) in which domain features have been provided in addition to a standard feature set. Our DL implementation considers only rules (features) for which

[11] *Senseval* is a contest for evaluating the strengths and weaknesses of WSD systems with respect to different words and different languages (see `http://www.senseval.org`).

an estimated statistical confidence is above a predefined threshold, following the methodology in (Dagan and Itai, 1994). Using decision lists it is possible to obtain quite accurate WSD classifiers, as described in (Martinez and Agirre, 2002), where the results obtained were just a few points worse than the best performing WSD systems on equivalent tasks (see (Preiss and Yarowsky, 2002)). It should be stressed though that our use of decision lists was intended to create a simple platform for testing and comparing the contribution of domain information in a clear and well controlled manner. While we do not attempt to re-produce here the best known WSD results, which were obtained by substantially more complex systems, we do hypothesize that the generic qualities of domains that are assessed by our experiments would be relevant for other supervised systems as well.

Decision lists have been used successfully to recognize collocational properties of sense distinctions, if trained using local (mostly syntagmatic) features, such as bigrams, trigrams and local context words. In many implementations *bag of words* (BOW) features have been also used to describe the broader (paradigmatic) context of the ambiguous word, as common in information retrieval style.

We compared two different feature sets for the supervised algorithm, creating two versions of the system:

BOW Local Features and Bag of Words

DOM Local Features and the domain labels D such that the $R(D, t) \geqslant k$, where t is the text window around the ambiguous word and k is a threshold, tuned empirically to 0.9 (recall that $R(D, t)$ is a probability value, providing a normalized scale of text relevance scores for all domains).

As local features, we use bigrams and trigrams of POS, lemmas and word forms that include the target word (as reported in Yarowsky (1994); Martinez and Agirre (2002)).

Both system versions consider syntagmatic as well as paradigmatic information. In BOW, paradigmatic information is represented by the standard bag of words feature set, while in DOM, paradigmatic information is represented using domain labels as features. Both feature sets were used in the lexical sample tasks, where supervised training data are available.

6.2 Evaluation of domain features in supervised WSD

This section reports experiments that support the following two claims within the supervised WSD setting:

(1) Using domain features yields better paradigmatic models, improving overall WSD performances.
(2) Using domain features requires fewer examples (than BOW) for generating paradigmatic models, improving the system's learning curve.

To asses these claims we evaluated the performance of DOM, BOW and supervised DDD on the Senseval-2 English lexical sample task. Concept vectors for the supervised DDD were learned from the available supervised training data as in Section 4.2.4.

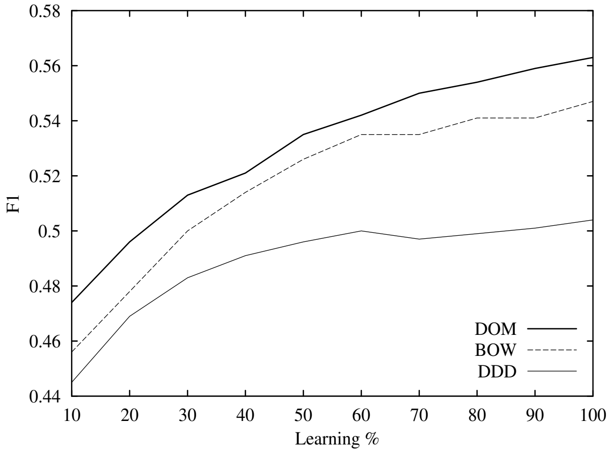

Fig. 2. Learning curves for LEX task (sense grained)

Figure 2 reports the learning curves for each of the systems. The figure shows that domain features improve the overall performance of DOM compared to BOW. Using full learning, DOM outperforms the performances of BOW by 2 point of F1 measure. Using domains as features also allows us to reduce the amount of annotations: DOM achieves the same performance as BOW with about 2/3 training (as mentioned earlier, using both domains and BOW features decreases performance). As foreseen, DDD achieves lower results if compared with both DOM and BOW, because it does not make use of syntagmatic information at all.

In addition we evaluated separately the performance on nouns and verbs, suspecting that nouns are more "domain oriented" than verbs. In this manner it is possible to test the hypothesis that domain information is more relevant for improving disambiguation of domain oriented words.

Figure 3 shows the learning curves of the previous systems on the same task evaluated separately for nouns and verbs. Domain information (DOM vs. BOW) contributes more for nouns (2% improvement for nouns vs. 1% for verbs) with full learning. In addition, DOM requires only 60% training to achieve the same performance as BOW does with full training. Supervised DDD is quite competitive for nouns compared to BOW. Its performance is better than BOW when using just 10% learning, and with full learning the

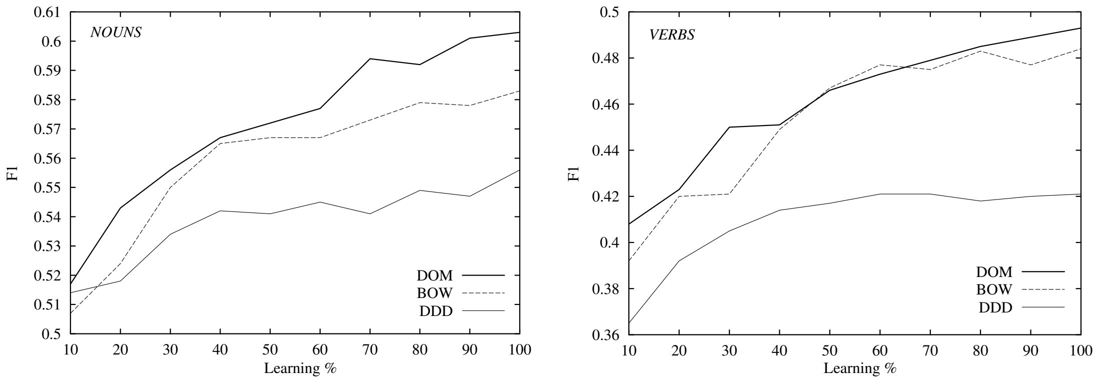

Fig. 3. Learning curves for LEX task (sense grained) - nouns (left) and verbs (right)

difference of accuracy is about 3%. On the other hand DDD is noticeably worse for verbs, for which it outperformed by BOW by 6 points. These results indicate the potential contribution of using domains in order to reduce annotation cost, in particular when taking into account that there are about three times more nouns than verbs in WordNet and that the amount of training examples available in the Senseval LEX test may not be realistic for the full range of nouns. Finally, these results confirm the hypothesis that nouns are more domain oriented than verbs.

The relative performance of domain features versus BOW has also been evaluated with respect to the potential of obtaining relatively high accuracy, possibly at lower coverage. Obtaining some reliable accurate classifications is an emerging topic, because accurate classifications can be used to generate automatically sense tagged examples, in order to train supervised systems (Yarowsky, 1995; Mihalcea, 2002).

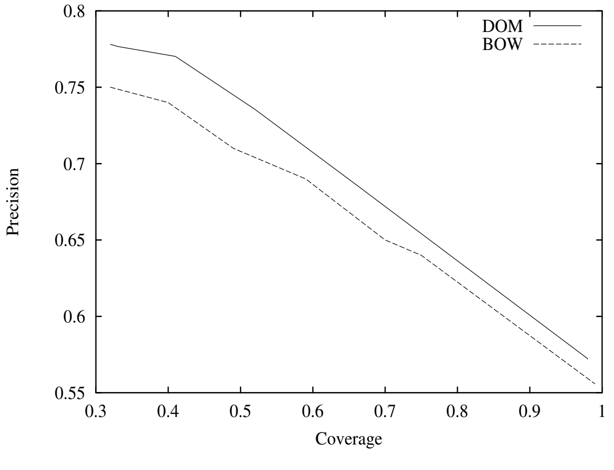

Fig. 4. Precision/Coverage curves for LEX task (sense grained)

Figure 4 compares the precision/coverage curves of DOM and BOW. DOM outperforms BOW at all points, and the divergence between the precision of the two systems increases with lower coverage values. The maximum preci-

	Precision	Attempted
Base Lesk	0.51	100
Most frequent	0.48	100

Table 3
Two baselines for LEX

sion of DOM is 0.78 at 0.3 coverage, 3 points higher than BOW at the same coverage.

To provide some comparative perspective for the above data, Table 3 reports the overall results of two supervised baselines for the LEX task (see (Preiss and Yarowsky, 2002)). The standard naive baseline is choosing consistently the most frequent sense for each word. In the Lesk algorithm, the most likely sense for a word in a given context is decided on a basis of a measure of contextual overlap between the current context and dictionary definitions available for the target word. In addition to dictionary definitions, Lesk algorithm uses manually annotated corpora, to augment the sense-centered context with additional tagged examples.

To conclude, these results demonstrate that domains provide informative and effective paradigmatic features to model sense distinctions. Using domain features improves both learning curves and overall performance relative to standard bag of words features. The advantage of using domains is more evident for nouns, confirming the original intuition that sense distinctions for nouns are more influenced by paradigmatic relations than for verbs. This result is also confirmed by the performance for nouns of the DDD system, which utilizes *only* domain information. The DDD results for nouns are closer to those obtained by a complete system (which utilizes both syntagmatic and paradigmatic features) than it is for verbs.

6.3 Domain Grained Evaluation

This section reports experiments testing the effectiveness of domain granularity sense distinctions for WSD. Sense annotations in the corpora were grouped by WordNet Domains information, considering as equivalent all the senses of a word that have the same domain annotation. Supervised DDD, DOM and BOW were trained on the Senseval-2 English Lexical Sample Task using domain distinctions instead of sense distinctions. A comparison of their learning curves is presented in Figure 5.

Most notably, domain level distinctions are indeed substantially easier to disambiguate than sense distinctions (max F1 of 0.80 vs. 0.56). Furthermore, the different systems do not get significantly better at the higher levels of learning,

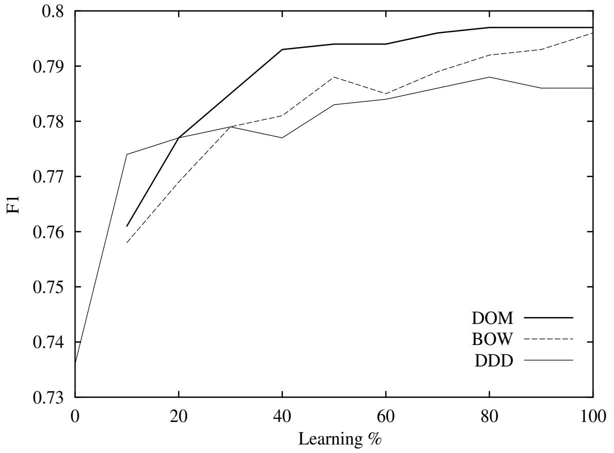

Fig. 5. Learning curves for LEX task (domain grained)

especially DOM and DDD, which utilize domain information. Both systems achieve their maximal F1 at 50% learning.

Figure 5 also indicates the F1 value for unsupervised DDD, which is the DDD result with 0% learning. The unsupervised F1 is about 5-6 points lower than the other supervised systems with full learning. This result is interesting as it shows that the DDD system can disambiguate all open class words in a text, including those for which training is not available, at a quite acceptable performance level from an application point of view.

An additional observation is that syntagmatic information is less relevant at domain level granularity. The DDD system, which makes use of only paradigmatic information, achieves a precision of 79%, which is only one point lower than the one of DOM, which makes use of both syntagmatic and paradigmatic information. In addition the learning curve of DDD is better until 20% of learning, demonstrating that DDD is appropriate to disambiguate domain level sense distinctions, in particular with little training.

6.4 Unsupervised Performance for All-words tasks

To assess further the use of domain information in unsupervised settings the DDD system (unsupervised version) was evaluated in the Senseval-2 all words task, using both sense and domain granularity level distinctions.

Figure 6 shows the precision/coverage curves of the system at both granularity levels. At full coverage the domain level precision is 0.83 compared to 0.62 for sense granularity. This result re-assesses (in the All-Words task) the practical effectiveness of DDD for domain level granularity in unsupervised settings. For sense granularity the DDD system achieve good precision levels only at

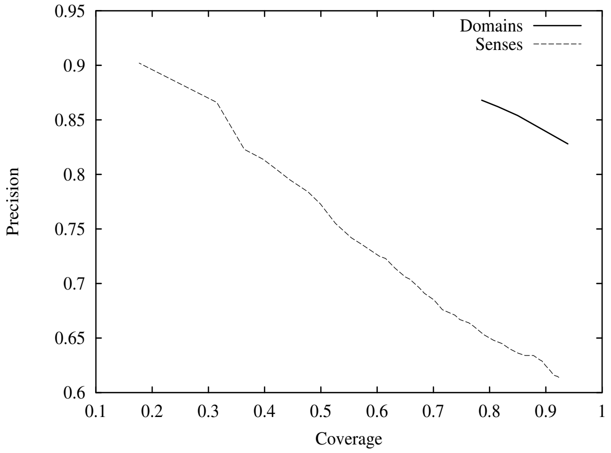

Fig. 6. Precision/Coverage curve for ALL (both domain and sense grained)

relatively low coverage (0.86 precision at 0.3 coverage).

The performance of unsupervised DDD for sense granularity was also evaluated separately for each POS. Results are reported in Table 4. As assumed, the system is not so accurate for verbs, but it achieves satisfactory results for the remaining POS. In particular the system achieves a good F1 figure for nouns, which is very close to state of the art results obtained by more complex supervised systems that use labeled training data (see (Preiss and Yarowsky, 2002) for a comparison among all systems on the Senseval-2 All Words task).

For comparison, Table 5 reports results for the standard "most frequent" baseline, as well as for two Senseval-2 All Words systems that exploit the notion of semantic domains, even though not necessarily using WordNet Domains information. Sinequa LIA-HMM exploits statistical models to identify some coarse semantic classes related to each word in the test, while DIMAP exploits the idea of topical area matches (see (Preiss and Yarowsky, 2002) and the Senseval web site for more information about these systems).

	Prec	Rec	Coverage	F1
nouns	0.626	0.613	0.980	0.620
verbs	0.437	0.434	0.993	0.435
adj	0.601	0.584	0.971	0.592
adv	0.756	0.753	0.996	0.754

Table 4
All-words sense grained results by POS

	Precision	Coverage
Sinequa LIA	0.61	1.0
DIMAP	0.45	1.0
Most frequent	0.605	1.0

Table 5
Other systems for All Words task that utilize the notion of semantic domains

7 Conclusions and Future Work

Domain analysis of text and lexicon can be utilized across different NLP tasks, allowing to unify certain representations and algorithms based on the same methodology and resources. Moreover, domains may constitute a bridge between the lexicon and the text, allowing a deeper comprehension within different lexical semantic phenomena at the paradigmatic level, such as ambiguity and lexical coherence. In this paper we have shown how domain information can be deduced in a principled unsupervised probabilistic manner based on the information available in WordNet Domains, yielding an effective generalized resource for word sense modeling.

In particular we considered two issues. First, domains provide informative generalized features for paradigmatic information that improve accuracy, or correspondingly - reduce the amount of annotated examples needed to obtain certain performance. This claim was assessed for WSD at both sense (synset) and domain levels of granularity.

Second, domain-level senses provide an appealing coarse granularity for WSD. Disambiguation at the domain level is substantially more accurate, while the accuracy of WSD for the fine-grained sense-level may not be good enough for various applications. Disambiguation at domain granularity is accurate and can be sufficiently practical using only domain information with the unsupervised DDD method alone, even with no training examples.

In future work we consider combining domain information with syntagmatic features in more sophisticated ways, relying on additional information from available lexical resources such as WordNet. Our eventual goal is to create a richer and largely unsupervised WSD solution, that might follow some of the underlying principles presented in this paper. We also plan to refine and enrich the domain annotation in WordNet Domains by developing corpus-based techniques for automatic acquisition of domain labels for synsets.

A Appendix: The Expectation Maximization Algorithm for the Gaussian Mixture Model

The framework provided by the Gaussian Mixture algorithm for Domain Relevance Estimation (described in 4.2.3) requires the definition of an algorithm to estimate the parameters to describe the required $PDFs$. The "empirical" PDF of domain frequency scores for each domain has to be decomposed into the weighted sum of two components, describing respectively relevant and non-relevant texts.

The Expectation Maximization (EM) algorithm for Gaussian Mixture (GM) Models (Redner and Walker, 1984) allows us to perform efficiently such operation. In this appendix details of this algorithm 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 A.1

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

with

$$a_j \geq 0 \text{ and } \sum_{j=1}^{m} a_j = 1 \tag{A.2}$$

and

$$G(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{A.3}$$

and $\theta =< a_1, \mu_1, \sigma_1, \ldots, a_m, \mu_m, \sigma_m >$ 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 A.4.

$$L(\mathcal{T}, D, \theta) = \prod_{t \in \mathcal{T}} p(F(D, t)|\theta) \tag{A.4}$$

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 A.5) is satisfied.

$$\theta_D = \underset{\theta'}{\operatorname{argmax}}\, L(\mathcal{T}, D, \theta') \tag{A.5}$$

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 us 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 achieves 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.

$$\Phi(\theta) = \Phi(< a_1, \mu_1, \sigma_1, a_2, \mu_2, \sigma_2 >) =< a'_1, \mu'_1, \sigma'_1, a'_2, \mu'_2, \sigma'_2 > \tag{A.6}$$

$$a'_j = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \frac{a_j G(F(D, t_k), \mu_j, \sigma_j)}{p(F(D, t_k), \theta)} \tag{A.7}$$

$$\mu'_j = \frac{\sum_{k=1}^{|\mathcal{T}|} F(D, t_k) \cdot \frac{a_j G(F(D,t_k),\mu_j,\sigma_j)}{p(F(D,t_k),\theta)}}{\sum_{k=1}^{|\mathcal{T}|} \frac{a_j G(F(D,t_k),\mu_j,\sigma_j)}{p(F(D,t_k),\theta)}} \tag{A.8}$$

$$\sigma'_j = \frac{\sum_{k=1}^{|\mathcal{T}|} (F(D, t_k) - \mu'_j)^2 \cdot \frac{a_i G(F(D,t_k),\mu_i,\sigma_i)}{p(F(D,t_k),\theta)}}{\sum_{k=1}^{|\mathcal{T}|} \frac{a_j G(F(D,t_k),\mu_j,\sigma_j)}{p(F(D,t_k),\theta)}} \tag{A.9}$$

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 3, with $c = 50$). 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 in all the experiments reported in this paper.

Acknowledgments

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

References

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

Comaroni, J. P., Beall, J., Matthews, W. E., New, G. R. (Eds.), 1989. Dewey Decimal Classification and Relative Index, 20th Edition. Forest Press, Albany, New York.

Dagan, I., Itai, A., 1994. Word sense disambiguation using a second language monolingual corpus. Computational Linguistics 20 (4), 536–596.

de Saussure, F., 1922. Cours de linguistique générale. Payot, Paris.

Escudero, G., Marquez, L., Rigau, G., 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In: Proceeding of Empirical Methods in Natural Language Processing 2000. Hong-Kong.

Fellbaum, C., 1998. WordNet. An Electronic Lexical Database. MIT Press.

Florian, R., Cucerzan, S., Schafer, C., Yarowsky, D., 2002. Combining classifiers for word sense disambiguation. Natural Language Engineering 8 (4), 327–341.

Gale, W., Church, K., Yarowsky, D., 1992. One sense per discourse. In: Proceedings of the 4^{th} DARPA Workshop on Speech and Natural Language Processing. Asilomar, California, pp. 233–237.

Gonzalo, J., Verdejio, F., Chugur, I., Cigarran, J., August 1998. Indexing with WordNet synsets can improve text retrieval. In: Harabagiu, S. (Ed.), Proceeding of the Workshop “Usage of WordNet in Natural Language Processing Systems”. Montreal, Quebec, Canada, pp. 38–44.

Guthrie, J., Guthrie, L., Wilks, Y., Aidinejad, H., June 1991. Subject-dependent co-occurence and word sense disambiguation. In: Proceedings of the 29^{th} Annual Meeting of the ACL. Berkeley, California, pp. 146–152.

Magnini, B., Cavaglià, G., June 2000. Integrating subject field codes into WordNet. In: Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation. Athens, Greece, pp. 1413–1418.

Magnini, B., Strapparava, C., October 2000. Experiments in word domain disambiguation for parallel texts. In: Proceedings of SIGLEX Workshop on Word Senses and Multi-linguality. Hong-Kong, pp. 27–33.

Magnini, B., Strapparava, C., July 2001. Improving user modelling with content-based techniques. In: UM2001 User Modeling: Proceedings of 8^{th} International Conference on User Modeling (UM2001). Springer Verlag, Sonthofen, Germany, pp. 74–83.

Magnini, B., Strapparava, C., Pezzulo, G., Gliozzo, A., July 2001. Using domain information for word sense disambiguation. In: Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation System. Toulose, France, pp. 111–114.

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

Martinez, D., Agirre, E., 2002. Syntactic features for high precision word sense disambiguation. In: Proceedings of the 19^{th} International Conference on Computational Linguistics (COLING). Taipei, Taiwan, pp. 626–632.

Mihalcea, R., May 2002. Bootstrapping large sense tagged corpora. In: Proceedings of the 3^{rd} International Conference on Languages Resources and Evaluations (LREC 2002). Las Palmas, Spain, pp. 1407–1411.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., Dang, H., July 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, pp. 21–24.

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

Procter, 1978. Longman Dictionary of Contemporary English.

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

Resnik, P., Yarowsky, D., April 1997. A perspective on word sense disambiguation methods and their evaluation. In: Light, M. (Ed.), Tagging Text with Lexical Semantics: Why, What and How? Washington, pp. 79–86, SIGLEX (Lexicon Special Interest Group) of the ACL.

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

Stevenson, M., Wilks, Y., 1999. Combining weak knowledge sources for sense disambiguation. In: Proceedings of International Joint Conference in Artificial Intelligence. Stockholm, Sweden, pp. 884–889.

Trier, J., 1934. Das sprachliche feld. eine auseinandersetzung. Neue Fachrbücher für Wissenschaft und Jugendbildung 10, 428–449.

Yarowsky, D., 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In: Proceedings of COLING92. Nantes, France, pp. 454–460.

Yarowsky, D., 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In: Proceedings of the 32^{nd} Annual Meeting of the ACL. Las Cruces, New Mexico, pp. 88–95.

Yarowsky, D., 1995. Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33^{rd} Annual Meeting of the ACL. Cambridge, MA, pp. 189–196.

Yarowsky, D., Florian, R., 2002. Evaluating sense disambiguation across diverse parameter space. Natural Language Engineering 8 (4), 293–310.