A New Minimally-Supervised Framework for Domain Word Sense Disambiguation

Stefano Faralli and **Roberto Navigli**
Dipartimento di Informatica
Sapienza Università di Roma
{faralli,navigli}@di.uniroma1.it

Abstract

We present a new minimally-supervised framework for performing domain-driven Word Sense Disambiguation (WSD). Glossaries for several domains are iteratively acquired from the Web by means of a bootstrapping technique. The acquired glosses are then used as the sense inventory for fully-unsupervised domain WSD. Our experiments, on new and gold-standard datasets, show that our wide-coverage framework enables high-performance results on dozens of domains at a coarse and fine-grained level.

1 Introduction

Domain information pervades most of the text we read every day. If we just think of the Web, the vast majority of its textual content is domain oriented. A case in point is Wikipedia, which provides encyclopedic coverage for a huge number of knowledge domains (Medelyan et al., 2009), but most blogs, Web sites and newspapers also provide a great deal of information focused on specific areas of knowledge. When it comes to automatic text understanding, then, it is crucial to take into account the domain specificity of a piece of text, so as to perform a focused and as-precise-as-possible analysis which, in its turn, can enable domain-aware applications such as question answering and information extraction. Domain knowledge also has the potential to improve open-text applications such as summarization (Ceylan et al., 2010) and machine translation (Foster et al., 2010).

Research in Word Sense Disambiguation (Navigli, 2009, WSD), the task aimed at the automatic labeling of text with word senses, has been oriented towards domain text understanding for several years now. Many approaches have been devised, including the identification of domain-specific predominant senses (McCarthy et al., 2007; Lapata and Keller, 2007), the development of domain resources (Magnini and Cavaglià, 2000; Magnini et al., 2002), their application to WSD (Gliozzo et al., 2004), and the effective use of link analysis algorithms such as Personalized PageRank (Agirre et al., 2009; Navigli et al., 2011). More recently, semi-supervised approaches to domain WSD have been proposed which aim at decreasing the amount of supervision needed to carry out the task (Khapra et al., 2010).

High-performance domain WSD, however, has been hampered by the widespread use of a general-purpose sense inventory, i.e., WordNet (Miller et al., 1990; Fellbaum, 1998). Unfortunately WordNet does not contain many specialized terms, making it difficult to use it in work on arbitrary specialized domains. While Wikipedia has recently been considered a valid alternative (Mihalcea, 2007), it is mainly focused on covering named entities and, strictly speaking, does not contain a formal wide-coverage sense inventory (not even in disambiguation pages, which are often incomplete, especially in the lexicographic sense).

In this paper we provide three main contributions:

- We tackle the above issues by introducing a new framework based on the minimally-supervised acquisition of specialized glossaries for dozens of domains.

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1411–1422, Jeju Island, Korea, 12–14 July 2012.

- In turn, we use the acquired domain glossaries as a sense inventory for domain WSD. As a result, we redefine the domain WSD task as one of picking out the most appropriate gloss (fine-grained setting) or domain (coarse-grained setting) from a multi-domain glossary.

- We show that our framework represents a considerable departure from the common usage of a general-purpose sense inventory such as WordNet, in that, thanks to the wide coverage of domain meanings, it enables high-performance unsupervised WSD on many domains in the range of 69-80% F1.

Furthermore, our approach can be customized to any set of domains of interest, and new senses, i.e., glosses, can be added at any time (either manually or automatically) to the multi-domain sense inventory.

2 Related Work

Domain WSD has been the focus of much interest in the last few years. An important research direction identifies distributionally similar neighbors in raw text as cues for determining the predominant sense of a target word by means of a semantic similarity measure (McCarthy et al., 2004; Koeling et al., 2005; McCarthy et al., 2007). Other distributional methods include the use of a word-category cooccurrence matrix, where categories are coarse senses obtained from an existing thesaurus (Mohammad and Hirst, 2006), and synonym-based word occurrence counts (Lapata and Keller, 2007). Domain-informed methods have also been proposed which make use of domain labels as cues for disambiguation purposes (Gliozzo et al., 2004).

Domain-driven approaches have been shown to obtain the best performance among the unsupervised alternatives (Strapparava et al., 2004), especially when domain kernels are coupled with a syntagmatic one (Gliozzo et al., 2005). However, their performance is typically lower than supervised systems. On the other hand, supervised systems fall short of carrying out high-performance WSD within domains, the main reason being the need for retraining on each new specific knowledge domain. Nonetheless, the knowledge acquisition bottleneck can be relieved by means of domain adaptation (Chan and Ng, 2006; Chan and Ng, 2007; Agirre and de Lacalle, 2009) or by effectively injecting a general-purpose corpus into a smaller domain-specific training set (Khapra et al., 2010).

However, as mentioned above, most work on domain WSD uses WordNet as a sense inventory. But even if WordNet senses have been enriched with topically-distinctive words and concepts (Agirre and de Lacalle, 2004; Cuadros and Rigau, 2008), manually-developed domain labels (Magnini et al., 2002), and disambiguated semantic relations (Navigli, 2005), the main obstacle of being stuck with an open-ended fine-grained sense inventory remains. Recent results on the SPORTS and FINANCE gold standard dataset (Koeling et al., 2005) show that domain WSD can achieve accuracy in the 50-60% ballpark when a state-of-the-art algorithm such as Personalized PageRank is paired with a distributional approach (Agirre et al., 2009) or with semantic model vectors acquired for many domains (Navigli et al., 2011).

In this paper, we take domain WSD to the next level by proposing a new framework based on the minimally-supervised acquisition of large domain sense inventories thanks to which high performance can be attained on virtually any domain using unsupervised algorithms. Glossary acquisition approaches in the literature are mostly focused on pattern-based definition extraction (Fujii and Ishikawa, 2000; Hovy et al., 2003; Fahmi and Bouma, 2006, among others) and lattice-based supervised models (Navigli and Velardi, 2010) starting from an initial terminology, while we jointly bootstrap the lexicon and the definitions for several domains with minimal supervision and without the requirement of domain-specific corpora. To do so, we adapt bootstrapping techniques (Brin, 1998; Agichtein and Gravano, 2000; Pasca et al., 2006) to the novel task of domain glossary acquisition from the Web.

Approaches to domain sense modeling have already been proposed which go beyond the WordNet sense inventory (Duan and Yates, 2010). Distinctive collocations are extracted from corpora and used as features to bootstrap a supervised WSD system. Experiments in the biomedical domain show good performance, however only in-domain ambiguity is addressed. In contrast, our approach tackles cross-

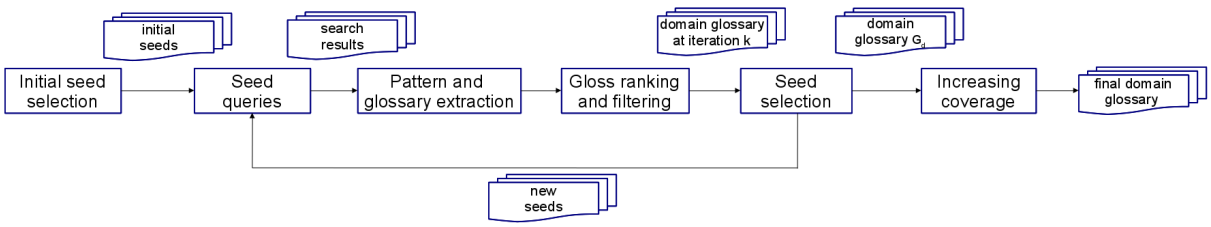

Figure 1: The bootstrapping process for glossary acquisition.

domain ambiguity, by working with virtually any set of domains and minimizing the requirements by harvesting domain terms and definitions from the Web, bootstrapped using a small number of seeds.

The existing approach closest to ours is that of Huang and Riloff (2010), who devised a bootstrapping approach to induce semantic class taggers from domain text. The semantic classes are associated with arbitrary NPs and must be established beforehand. Our objective, instead, is to perform domain disambiguation at the word level. To do this, we redefine the domain WSD problem as one of selecting the most suitable gloss from those available in our full-fledged multi-domain glossary.

3 A Minimally-Supervised Framework for Domain WSD

In this section we present our new framework for performing domain WSD. The framework consists of two phases: glossary bootstrapping (Section 3.1) and domain WSD (Section 3.2).

3.1 Phase 1: Bootstrapping Domain Glossaries

The objective of the first phase is to acquire a multi-domain glossary from the Web with minimal supervision. We initially select a set D of domains of interest. For each individual domain $d \in D$ we start with an empty set of HTML patterns P_d (i.e., $P_d := \emptyset$), used for gloss harvesting. During this phase we iteratively populate the pattern set by means of six steps, described in the next six subsections and depicted in Figure 1. The final output of this phase will be a glossary G_d consisting of domain terms and their automatically-harvested glosses.

3.1.1 Step 1: Initial seed selection

First, given the domain d, we manually pick out K hypernymy relation seeds $S_d = \{(t_1, h_1), \ldots, (t_K, h_K)\}$, where the pair ($t_i$, h_i) contains a domain term t_i and its generalization h_i (e.g., (*firewall*, *security system*)). The only constraint we impose is that the selected relations must be distinctive for the domain d of interest. The chosen hypernymy relations have to be as topical and representative as possible for the given domain (e.g., (*compiler*, *computer program*) is an appropriate pair for computer science, while (*byte*, *unit of measurement*) is not, as it might cause the extraction of several glossaries of various units and measures). Note that this is the only human intervention in the entire glossary acquisition process.

We now set the iteration counter k to 1 and start the first iteration of the process (steps 2-5). After each iteration k, we keep track of the set of glosses G_d^k, acquired during iteration k.

3.1.2 Step 2: Seed queries

For each seed pair (t_i, h_i), we submit the following three queries to a Web search engine: "t_i" "h_i" `glossary`[1], "t_i" "h_i" `definition`, "t_i" "h_i" `dictionary` and collect the 64 top-ranking results for each query[2]. Each resulting page is a candidate glossary for the domain d identified by our relation seeds S_d.

3.1.3 Step 3: Pattern and glossary extraction

We initialize the glossary for iteration k as follows: $G_d^k := \emptyset$. Next, from each resulting page, we harvest all the text snippets s starting with t_i and ending with h_i (e.g., *firewall*`</b>` `--` a `<i>`*security system*), i.e., $s = t_i \ldots h_i$. For each such text snippet s, we perform five substeps:

a) **extraction of the term/gloss separator:** we

[1] In what follows, we use the `typewriter font` for keywords and term/gloss separators.

[2] We use the Google AJAX API, which returns 64 results.

Term	Gloss	Hypernym	# seeds	Gloss score
dynamic packet filter	A **firewall** facility that monitors the state of connections and uses this information to determine which network packets to allow through the **firewall**	firewall	2	0.75
peripheral	Hardware that extends the capabilities of the computer, such as a printer, modem, or **scanner**.	hardware	1	0.83
die	An integrated circuit **chip** cut from a finished wafer.	integrated circuit	1	0.75
constructor	a method used to help create a new object and initialise its data	method	0	1.00
schema	In database terminology, a schema is the organization of the tables, the fields in each table, and the relationships between fields and tables.	database	0	0.78

Table 1: Examples of extracted terms, glosses and hypernyms (seeds are in bold, domain terms are underlined).

start from t_i and move right until we extract the longest sequence p_M of HTML tags and non-alphanumeric characters, which we call the *term/gloss separator*, between t_i and the glossary definition (e.g., " --" between "firewall" and "a" in the above example);

b) **gloss extraction:** we expand the snippet s to the right of h_i in search of the entire gloss of t_i, i.e., until we reach a non-formatting tag element (e.g., , ,), while ignoring formatting elements such as , and <a> which are typically included within a definition sentence. As a result, we obtain the sequence $t_i\ p_M\ gloss_s(t_i)\ p_R$, where $gloss_s(t_i)$ is our gloss for seed term t_i in snippet s (which includes h_i by construction) and p_R is the non-formatting HTML tag element to the right of the extracted gloss. For example, we extend the above definition for *firewall* to: "a *security system* for protecting against illegal entry to a local area network.".

c) **pattern instance extraction:** we extract the following pattern instance:

$$p_L\ t_i\ p_M\ gloss_s(t_i)\ p_R,$$

where p_L and p_R are, respectively, the left boundary of t_i and the right boundary of $gloss_s(t_i)$, and p_M is the term/gloss separator extracted at step 3(a). The two boundaries p_L and p_R are obtained by extracting the longest sequence of HTML tags and non-alphanumeric characters obtained when moving to the left of t_i and the right of $gloss_s(t_i)$, respectively[3]. For the above example, we extract the following pattern instance: p_L = " ", t_i = "*firewall*", p_M = " --", $gloss_s(t_i)$ = "a *security system* for protecting against illegal entry to a local area network.", p_R =" ".

d) **pattern extraction:** we generalize the above pattern instance to the following pattern:

$$p_L\ *\ p_M\ *\ p_R,$$

i.e., we replace t_i and $gloss_s(t_i)$ with *. In the above example, we obtain the following pattern:

 * -- * .

Finally, we add the generalized pattern to the set of patterns P_d, i.e., we set $P_d := P_d \cup \{p_L\ *\ p_M\ *\ p_R\}$. We also add the first sentence of the retrieved definition $gloss_s(t_i)$ to our glossary G_d^k, i.e., $G_d^k := G_d^k \cup \{(t_i, first(gloss_s(t_i)))\}$, where $first(g)$ returns the first sentence of gloss g.

e) **pattern matching:** we look for additional pairs of terms/glosses in the Web page containing the snippet s by matching the page against the generalized pattern $p_L\ *\ p_M\ *\ p_R$. We then add to G_d^k the new (term, gloss) pairs matching the generalized pattern.

As a result of this step, we obtain a glossary G_d^k for the terms discovered at iteration k.

3.1.4 Step 4: Gloss ranking and filtering

Importantly not all the extracted definitions pertain to the domain of interest. In order to rank by domain pertinence the glosses obtained at iteration k, we define the terminology T_1^{k-1} of the terms accumulated up until iteration $k-1$ as follows: $T_1^{k-1} := \bigcup_{i=1}^{k-1} T^i$, where $T^i := \{t : \exists(t,g) \in G_d^i\}$.

[3]The minimum and maximum length of both p_L and p_R are set to 4 and 50 characters, respectively, as a result of a tuning phase described in Section 4.1.

Gloss	Domain
Measures undertaken to return a degraded ecosystem's functions and values, including its hydrology, plant and...	BIOLOGY
The renewing or repairing of a natural system so that its functions and qualities are comparable to its original...	GEOGRAPHY
The reign of Charles II in England.	ROYALTY
A goal of criminal sentencing that attempts to make the victim "whole again."	LAW
The process and work of improving the degraded quality of the sound or image in terms of video and audio preservation.	MEDIA
A process used by radio astronomers to eliminate the smoothing effect observed in radio maps that is caused by...	PHYSICS

Table 2: Examples of glosses harvested for the term *restoration*.

For the base step $k = 1$, we define $T_1^0 := T^1$, i.e., we use the first-iteration terminology itself. To rank the glosses, we first transform each acquired gloss g to its bag-of-words representation $Bag(g)$, which contains all the single- and multi-word expressions in g. We then score each gloss g by the ratio of domain terms found in its bag of words:

$$score(g) = \frac{|Bag(g) \cap T_1^{k-1}|}{|Bag(g)|}. \quad (1)$$

In Table 1 we show some glosses in the computer science domain (second column, domain terms are underlined) together with their score (last column). Next, we use a threshold θ (tuned on a held-out domain, described in Section 4.1) to remove from G_d^k those glosses g whose $score(g) < \theta$.

3.1.5 Step 5: Seed selection for next iteration

We now aim at selecting the new set of hypernymy relation seeds to be used to start the next iteration. We perform three substeps:

a) **Hypernym extraction:** for each newly-acquired term/gloss pair $(t, g) \in G_d^k$, we automatically extract a candidate hypernym h from the textual gloss g. To do this we use a simple unsupervised heuristic which just selects the first term in the gloss. More sophisticated, supervised approaches could have been used for hypernym extraction from glosses (Navigli and Velardi, 2010). However, note that, for the purposes of our glossary extraction task, it is not crucial to extract accurate hypernyms, but rather to harvest terms h which are very likely to occur in the glosses of t. We show an example of hypernym extraction for some terms in Table 1 (we report the term in column 1, the gloss in column 2 and the hypernyms extracted by our hypernym extraction technique in column 3).

b) **(Term, Hypernym)-ranking:** we sort all the glosses in G_d^k by the number of seed terms found in each gloss. In the case of ties (i.e., glosses with the same number of seed terms), we further sort the glosses by the score shown in Formula 1. We show the number of seed terms and the scores for some glosses in Table 1 (columns 4 and 5, respectively), where seed terms are in bold and domain terms (i.e., in T_1^{k-1}) are underlined.

c) **New seed selection:** as new seeds we select the (term, hypernym) pairs corresponding to the K top-ranking glosses.

If k equals the maximum number of iterations, we stop. Else, we increment the iteration counter (i.e., $k := k + 1$) and jump to step (2) of our glossary bootstrapping algorithm after replacing S_d with the new set of seeds.

The output of the glossary bootstrapping phase is a domain glossary $G_d := \bigcup_{i=1,...,max} G_d^i$, where max is the total number of iterations.

3.1.6 Step 6: Increasing Coverage

Given the nature of Web domain glossaries one can rarely find terms and definitions for general terms (e.g., *jurisprudence* for the LAW domain). In order to cover this gap, we apply domain filtering (see Section 3.1.4) to all the glosses contained in a general-purpose dictionary (we use WordNet). We then add the surviving term/gloss pairs to G_d.

3.2 Phase 2: Domain WSD

Now that we have acquired a glossary for each domain in our set D, we can create a multi-domain glossary $\mathcal{G} := \{((t, g), d) : d \in D, (t, g) \in G_d\}$. Our glossary $\mathcal{G}$ is thus a set of term/gloss pairs for many domains. Note that one pair might individually belong to more than one domain, as glossary bootstrapping is performed separately for each domain. In Table 2 we show an example of the

glosses acquired for the term *restoration*. We observe that 5 out of 6 senses are not available in WordNet (namely: the BIOLOGY, GEOGRAPHY, LAW, MEDIA and PHYSICS senses). Many of them are domain-specific meanings for the general concept of "the act of restoring", with the BIOLOGY and GEOGRAPHY senses being very similar. However, this is a perfectly acceptable phenomenon as any of the two senses, i.e., glosses, would be equally valid when disambiguating a domain text dealing with ecosystem restoration.

3.2.1 Gloss-driven WSD

We redefine the task of domain WSD as one of selecting the most suitable gloss, if one exists, for an input term t. For instance, consider the sentence: "He performed the *restoration* of heavily corrupted images". An appropriate option for this occurrence would be the MEDIA sense of *restoration* in Table 2.

Our gloss-driven WSD paradigm has the desirable property of automatically providing two levels of sense granularity: a domain, coarse-grained level, similar in spirit to Word Domain Disambiguation (Sanfilippo et al., 2006), in which the sense inventory of a term t is just the set of domains for which t is covered (e.g., BIOLOGY, GEOGRAPHY, ROYALTY, LAW, MEDIA, PHYSICS in the example of Table 2), and a fine-grained level, which requires the selection of the gloss which best describes the sense denoted by the given word occurrence. A second desirable property of our gloss-driven WSD paradigm is that it relies on a flexible framework, which allows for the bootstrapping of new domain glossaries or the expansion of existing ones. However, while these two properties – i.e., double level of granularity distinctions and flexibility – are naturally inherent in the gloss-driven paradigm, the same cannot be said for mainstream open-text WSD in which general-purpose static dictionaries are typically used.

In order to evaluate our framework for domain WSD, we propose two fully unsupervised algorithms for gloss-driven domain WSD. Ideally, high performance could be obtained using state-of-the-art supervised WSD systems. However, in order to train such systems, a wide-coverage sense-labeled corpus should be available for each domain, a heavy task which we leave to future work. Instead, our objective is to show that high-performance domain WSD can be enabled with little effort by our framework.

3.2.2 Algorithm 1: WSD with Personalized PageRank

Domain Glossaries as Graphs For each domain $d \in D$, we create an undirected graph $N_d = (V_d, E_d)$ as follows: V_d is the set of concepts identified by term/gloss pairs in the domain glossary G_d, i.e., $V_d := G_d$; E_d is the set of edges between pairs of concepts, where an edge $\{(t, g), (t', g')\}$ exists if and only if t' is such that $t' \neq t$ and t' occurs in the bag of words of the gloss g of t. In other words, t is connected to all the domain senses of words used in its definition g.

Graph-based WSD Given an input text, for each domain $d \in D$, we produce its bag of domain content words $C_d = \{w_1, w_2, \ldots, w_n\}$ by performing tokenization, lemmatization and compounding based on the lexicon of domain d. Then, given a target word t, we use $C_d \setminus \{t\}$ as the context to disambiguate t within the domain d. In order to carry out domain WSD, i.e., to pick out the most suitable sense of t across domains, we apply a state-of-the-art graph-based algorithm, namely Personalized PageRank (Haveliwala, 2002, PPR), to each domain graph N_d. PPR is a variant of the popular PageRank algorithm (Brin and Page, 1998) in which the damping probability mass is concentrated on a selected number of graph nodes, instead of being uniformly distributed across all nodes. Specifically, following Agirre and Soroa (2009) we concentrate the probability mass on the nodes $(t', g') \in V_d$ for which the term t' is a context word, i.e., $t' \in C_d$. Next, for each domain $d \in D$, we run PPR for a given number of iterations and obtain as output a probability distribution PPV_d over the graph nodes. Finally, we select the most suitable gloss of t as follows:

$$Sense_{PPR}(t) = \underset{g:\exists d \in D,(t,g) \in V_d}{\arg\max} PPV_d(t, g) \quad (2)$$

where $PPV_d(t, g)$ is the PPR probability for the term/gloss pair (t, g) and $Sense_{PPR}(t)$ contains the best interpretation of t across all the domains D.

3.2.3 Algorithm 2: PPR Boosted with Domain Distribution Information

The words in a given text do not typically deal with a single domain. Instead, they touch different

ART	BIOLOGY	BUSINESS	CHEMISTRY	COMPUTING	EDUCATION	ENGINEERING	ENVIRONMENT	FOOD & DRINK	GEOGRAPHY
GEOLOGY	HEALTH	HISTORY	LANGUAGE	LAW	LITERATURE	MATHS	MEDIA	METEOROLOGY	MUSIC
PHILOSOPHY	PHYSICS	POLITICS	PSYCHOLOGY	RELIGION	ROYALTY	SPORTS	TOURISM	VIDEOGAMES	WARFARE

Table 3: List of the 30 domains used in our experiments.

COMPUTING		FOOD		ENVIRONMENT		BUSINESS	
chip	circuit	timbale	dish	sewage	waste	eurobond	bond
destructor	method	brioche	bread	acid rain	rain	asset play	stock
compiler	program	macaroni	pasta	ecosystem	system	income stock	security
html	language	pizza	dish	air monitoring	sampling	financial intermediary	institution
firewall	security system	ice cream	dessert	global warming	temperature	derivative	financial product
remote lan access	process	pasteurized milk	milk	fermentation	decomposition	arbitrage pricing theory	economic theory
relational database	tabular database	salted butter	butter	attainment area	area	banker's draft	bill of exchange
admin console	user interface	prosecco	wine	fugitive dust	matter	working capital	cash

Table 4: Hypernymy relation seeds used to bootstrap glossary acquisition in four of the 30 domains.

areas of knowledge which are intertwined with each other within the discourse. For example, a text dealing with VIDEOGAMES will often concern domains such as BUSINESS, COMPUTING, SPORTS, etc. Given an input text, we can capture its relevance for each domain by calculating the following domain score:

$$\beta_d = \frac{|C_d|}{\sum_{d' \in D} |C_{d'}|} \quad (3)$$

where, as above, C_d is the set of content words from the input text which are covered by domain d. We thus propose a second algorithm which synergistically combines the spreading effect of PPR with the domain distribution information. The best sense for a given term t is calculated as follows:

$$Sense_{DomPPR}(t) = \underset{g:\exists d \in D,(t,g) \in V_d}{\arg\max} \beta_d PPV_d(t,g) \quad (4)$$

that is, we select as the most suitable gloss for t the one which maximizes the product of its domain relevance score by its domain PPV_d value. Note that the same gloss can occur in multiple domains and that it might obtain different scores depending on the domain. Again, since the approach is gloss-driven, we do not see this as a problem, but rather as a natural characteristic of our framework.

4 Experimental Setup

4.1 Domains

We selected 30 domains starting from the Wikipedia featured articles[4]. We show the domain labels in Table 3 (some labels have been conveniently shortened, e.g., PHYSICS should read PHYSICS & ASTRONOMY). We manually identified 8 hypernym/hyponym seeds for each domain, totalizing 240 seeds. We used two criteria for selecting a seed: i) it covers a separate segment of the domain, and ii) it has to be specialized enough to avoid ambiguity. We show the seeds used in four of our domains in Table 4. We bootstrapped our glossary acquisition technique (cf. Section 3.1) on each domain and performed 5 iterations. For increasing the coverage of domain terms we used WordNet glosses (see Section 3.1.6). As a result, we obtained 30 domain glossaries. We also kept aside a 31st domain, namely FASHION, which we employed for tuning the minimum and maximum length of both p_L and p_R in Section 3.1.3 and the threshold θ used to filter out non-domain glosses in Section 3.1.4.

Table 5: Statistics on the multi-domain acquired glossary.

	From the Web	From WordNet	From both	Total
Terms	74,295	83,904	18,313	176,512
Glosses	153,920	68,731	596	223,247

In Table 5 we show the statistics for the acquired multi-domain glossary by distinguishing Web-derived and WordNet terms and glosses.

4.2 Sense Inventory

Our sense inventory is given by the 30-domain glossary obtained as a result of our glossary bootstrapping phase. Overall we collected 176,512 and 223,247 distinct terms and glosses, respectively, with an important contribution from both the Web

[4] http://en.wikipedia.org/wiki/Wikipedia:Featured_articles

and WordNet (see Table 5). The average number of glosses per term in our inventory is 1.9 (3.6 glosses on polysemous terms). However, note that a monosemous word in our domain sense inventory does not necessarily make disambiguation easier, as i) we might have missed other domain-specific senses, ii) an uncovered, non-domain sense might fit a word occurrence (in this case, the domain WSD algorithms might be (wrongly) biased towards returning the only possible choice if a non-zero disambiguation score is calculated for it).

In order to determine the suitability of our multi-domain sense inventory, we compared it with the latest version of WordNet Domains (Magnini et al., 2002, WND 3.2), a well-known resource which provides domain labels for almost 65,000 nominal WordNet synsets (we removed all the synsets tagged with the FACTOTUM label, which indicates no domain specificity). Since WND uses about 160 finer-grained domain labels, we manually mapped them to our 30 labels when possible (e.g. SOCCER and SWIMMING were mapped to SPORTS), totalizing 62,100 domain-labeled synsets.

We calculated the coverage of our sense inventory against WND at the synset and the sense level, for each non-FACTOTUM synset. Given a WordNet synset S, let $d = \bigcup_{s \in S} d_s$ be the union of the domains d_s provided for each synonym $s \in S$ by our sense inventory ($d_s = \emptyset$ if not present), and let d' be the domain labels assigned to S by WND. A synset is covered if d and d' intersect. At the sense level, instead, we consider a synonym $s \in S$ to be covered if d_s and d' intersect. Our synset and sense coverage is 65.9% (40,969/62,100) and 63.7% (71,950/112,875), respectively. We also calculated an extra-coverage of 203.2% (229,384/112,875), that is the fraction of domain senses which are not available in WND, but we are able to provide in our sense inventory (see e.g. the example in Table 2) over the total number of senses in WND. While coverage and extra-coverage provide a good indicator of the completeness of our sense inventory, we need to calculate its precision to determine its correctness. To do so, we randomly sampled 500 domain glosses of terms for which no WordNet sense was tagged with the same domain in WND. A manual validation of this sample resulted in an 87.0% (435/500) estimate of the precision of our sense inventory.

4.3 Datasets

A dataset for 30 domains We used the Gigaword corpus (Graff and Cieri, 2003) to extract a 6-paragraph text snippet for each of the 30 domains. As a result, we obtained a domain dataset made up of 180 paragraphs to which we applied tokenization, lemmatization and compounding, totaling 1432 domain content words overall (47.7 content words per domain on average). The average polysemy of the words in the dataset was of 9.7 glosses and 4.4 domains per word. Each content word was manually tagged with the most suitable glosses from our multi-domain glossary (3.9 glosses, i.e., senses per word were assigned on average). The annotation task was performed by two annotators with adjudication.

Sports and Finance We also experimented with the gold standard produced by Koeling et al. (2005). The dataset covers two domains: SPORTS and FINANCE. The dataset comprises 41 ambiguous words (with an average polysemy of 6.7 senses), many of which express different meanings in the two domains. In each domain, and for each word, around 100 sentences were sense-annotated with WordNet.

Environment Finally, we also carried out an experiment on the ENVIRONMENT dataset from the Semeval-2010 domain WSD task (Agirre et al., 2010). The dataset includes 1,398 content words (of which 1,032 content nouns) tagged with WordNet senses.

4.4 Systems

We applied the two algorithms proposed in Section 3.2, namely vanilla PPR and domain-boosted PPR. For both versions of PPR we employed UKB, a readily-available implementation of PPR for WSD[5], successfully experimented by Agirre and Soroa (2009) and Agirre et al. (2009).

4.5 Baselines

Random baseline We compared our algorithms with the random baseline, which associates a random gloss among those available for each word occurrence according to a uniform distribution.

[5] http://ixa2.si.ehu.es/ukb/

Predominant domain We also compared our algorithms with a predominant sense baseline which assigns to each word occurrence the domain label with the highest domain score β_d among those available for the word (cf. Formula 3). Note that this is a strong baseline, because it aims at identifying the domain covered by the majority of terms in the input text, however it can disambiguate only at a coarse-grained level, i.e., at the domain level.

5 Experimental Results

30 domains We ran our WSD systems and the baselines on our 30-domain dataset, on a sentence-by-sentence basis. We calculated results at the two levels of granularity enabled by our WSD framework: a coarse-grained setting where systems output the most appropriate domain label for each word item to be disambiguated; a fine-grained setting where systems are required to output the most suitable gloss for the input word. The results are shown in Table 6. Domain PPR outperforms Vanilla PPR by some points in precision, recall and F1 in both the coarse-grained and the fine-grained setting, achieving an F1 around 80% and 69%, respectively (differences in recall performance are statistically significant using a χ^2 test). The predominant domain baseline, available only in the coarse-grained setting, lags behind Domain PPR by more than 3 points in precision and 2 in recall. While these differences are not statistically significant, the variance across domains is much higher, thus suggesting lower reliability of the method.

These results were obtained in a fully unsupervised setting in which no structured knowledge was provided, unlike previous applications of PPR to WSD (Agirre et al., 2009; Agirre and Soroa, 2009) which relied on the underlying WordNet graph, a manually created resource. Furthermore, our graph contains "noisy" semantic relations, as we connect each gloss to all the senses of its gloss words (cf. Section 3.2.2). Finally, we note that the results shown in Table 6 could never have been obtained with WordNet. In fact, drawing on our domain mapping, we calculated that the correct domain sense is not in WordNet for about 68% of the words in the dataset. Instead, the results in Table 6 show that our framework enables high-performance unsupervised WSD thanks to the wide coverage of domain meanings.

	Coarse-grained			Fine-grained		
	P	R	F1	P	R	F1
Vanilla PPR	76.7	74.3†	75.5	66.1	64.1†	65.1
Domain PPR	81.2	78.7†	79.9	69.7	67.6†	68.6
Predom. domain	77.9	76.8	77.3	-	-	-
Random baseline	42.5	42.5	42.5	44.1	44.1	44.1

Table 6: Performance results on the 30-domain dataset († differences between the two systems are statistically significant using a χ^2 test, $p < 0.05$).

As regards the random baseline, this performs 42.5% and 44.1% in the two settings. Despite the higher polysemy of glosses (9.7 glosses vs. 4.4 domains per word in the dataset), the performance is higher in the fine-grained setting because often there is more than one gloss covering the same meaning of a domain word.

Sports, Finance and Environment For the SPORTS, FINANCE and ENVIRONMENT datasets (cf. Section 4.3) we did not have gloss-based sense annotations, so we could not perform a fine-grained evaluation. Therefore, we first studied the different systems at a coarse level on the basis of the domain distribution of the senses returned for the word items in the dataset. We show the 3 most frequent domain labels for each system and each dataset in Figure 2. The figure seems to confirm our results showing Domain PPR as being more robust than its Vanilla version. Next, to get a more accurate evaluation, we randomly sampled 200 sentences from each dataset and manually validated the coarse-grained senses, i.e., domain assignments, output by each system on this set of sentences. We remark that several words in the datasets did not pertain to the domain of interest. For instance, *will* and *share* do not have any sports sense in WordNet, while the same applies to *half* and *chip* for the business domain. Table 7 shows the results of our validation, where a domain output by a system was considered correct if a suitable gloss existed for that domain in our inventory.

The results show that our framework enables coarse-grained recall in the 70-80% ballpark even on difficult gold standard datasets for which fine-grained recall with WordNet struggles to surpass the 50-60% range. For instance, the best performance

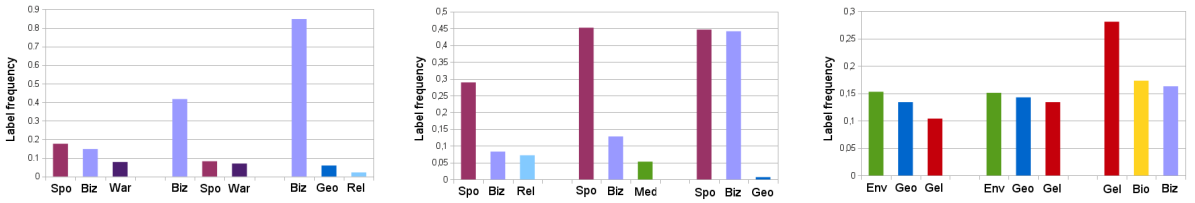

Figure 2: Frequency of the most common domain labels returned by our 3 systems on standard domain datasets.

	FINANCE			SPORTS			ENVIRONMENT		
	P	R	F1	P	R	F1	P	R	F1
Vanilla PPR	57.8	56.5	57.1	65.5	63.2	64.3	81.5	77.9	79.7
Domain PPR	77.8	76.1	76.9	72.1	71.3	71.7	83.1	79.4	81.2
Predom. domain	80.0	78.3	79.1	72.6	70.1	71.3	72.7	70.6	71.6

Table 7: Coarse-grained performance results on gold-standard domain datasets.

on the ENVIRONMENT dataset was around 60% recall (Kulkarni et al., 2010) using a semi-supervised WSD system, trained on the domain. Similarly, both the FINANCE and SPORTS datasets are notoriously difficult gold standards on which state-of-the-art recall using WordNet is lower than 60% (Navigli et al., 2011).

Interestingly, the predominant domain baseline shows a bias towards BUSINESS, thus performing best on the FINANCE dataset. This is because of the large number of terms covered in our domain glossary, and consequently the high overlap with cue words in context. On the other two domains, we observe performance in line with our 30-domain experiment.

6 Conclusion

We have here presented a new framework for domain Word Sense Disambiguation. We depart from the use of general-purpose sense inventories like WordNet and propose a bootstrapping approach to the acquisition of sense inventories for virtually any domain. While we selected 30 domains for this study, nothing would prevent us from using a smaller or larger set of these domains, or a set of completely different domains.

Our work provides three main contributions:

i) we propose a new, flexible approach to glossary bootstrapping which harvests hundreds of thousands of term/gloss pairs; the resulting multi-domain glossary is shown to have wide coverage across domains and to include a large amount of terms not available in WordNet;

ii) we propose a novel framework for fully-unsupervised domain WSD which uses the multi-domain glossary as our sense inventory;

iii) we show that high performance can be achieved by means of simple, unsupervised WSD algorithms (around 80% and 69% in a coarse- and fine-grained setting, respectively).

Note that our aim here has not been to determine which system performs best, but rather to show that a reliable, full-fledged framework for domain WSD can be set up with minimal supervision. Additionally, our framework can be applied to any language of interest, provided enough glossaries are available online, by simply translating the keywords used for our queries.

The multi-domain glossary (and sense inventory) together with the seeds used for bootstrapping are available from `http://lcl.uniroma1.it/dwsd`.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital Libraries (DL 2000)*, pages 85–94, San Antonio, Texas, United States.

Eneko Agirre and Oier Lopez de Lacalle. 2004. Publicly available topic signatures for all WordNet nominal senses. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, LREC 2004, pages 1123–1126, Lisbon, Portugal.

Eneko Agirre and Oier Lopez de Lacalle. 2009. Supervised domain adaption for WSD. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2009, pages 42–50, Athens, Greece.

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2009, pages 33–41, Athens, Greece.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2009. Knowledge-based WSD on specific domains: performing better than generic supervised WSD. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1501–1506, Pasadena, California.

Eneko Agirre, Oier López de Lacalle, Christiane Fellbaum, Shu-Kai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers. 2010. Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 75–80, Uppsala, Sweden.

Sergey Brin and Michael Page. 1998. Anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7^{th} Conference on World Wide Web*, WWW 2007, pages 107–117, Brisbane, Australia.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *Proceedings of the International Workshop on The World Wide Web and Databases (WebDB 1998)*, pages 172–183, London, UK.

Hakan Ceylan, Rada Mihalcea, Umut Özertem, Elena Lloret, and Manuel Palomar. 2010. Quantifying the limits and success of extractive summarization systems across domains. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 903–911, Los Angeles, California.

Yee Seng Chan and Hwee Tou Ng. 2006. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL 2006, pages 89–96, Sydney, Australia.

Yee Seng Chan and Hwee Tou Ng. 2007. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL 2007, pages 49–56, Prague, Czech Republic.

Montse Cuadros and German Rigau. 2008. KnowNet: building a large net of knowledge from the Web. In *Proceedings of the 22nd International Conference on Computational Linguistics*, COLING 2008, pages 161–168, Manchester, U.K.

Weisi Duan and Alexander Yates. 2010. Extracting glosses to disambiguate word senses. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, NAACL 2010, pages 627–635, Los Angeles, California, USA.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications*, pages 64–71, Trento, Italy.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 451–459, Cambridge, Massachusetts.

Atsushi Fujii and Tetsuya Ishikawa. 2000. Utilizing the world wide web as an encyclopedia: extracting term descriptions from semi-structured texts. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL 2000, pages 488–495, Hong Kong.

Alfio Gliozzo, Carlo Strapparava, and Ido Dagan. 2004. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech and Language*, 18(3):275–299.

Alfio Gliozzo, Claudio Giuliano, and Carlo Strapparava. 2005. Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL 2005, pages 403–410, Ann Arbor, Michigan.

David Graff and Christopher Cieri. 2003. English Gigaword, LDC2003T05. In *Linguistic Data Consortium*, Philadelphia.

Taher H. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of 11th International Conference on World Wide Web*, WWW 2002, pages 517–526, Honolulu, Hawaii.

Eduard Hovy, Andrew Philpot, Judith Klavans, Ulrich Germann, and Peter T. Davis. 2003. Extending metadata definitions by automatically extracting and organizing glossary definitions. In *Proceedings of the 2003 Annual National Conference on Digital Government Research*, pages 1–6, Boston, MA.

Ruihong Huang and Ellen Riloff. 2010. Inducing domain-specific semantic class taggers from (almost) nothing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL 2010, pages 275–285, Uppsala, Sweden.

Mitesh Khapra, Anup Kulkarni, Saurabh Sohoney, and Pushpak Bhattacharyya. 2010. All words domain adapted WSD: Finding a middle ground between supervision and unsupervision. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1532–1541, Sweden.

Rob Koeling, Diana McCarthy, and John Carroll. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing*, pages 419–426, Vancouver, B.C., Canada.

Anup Kulkarni, Mitesh Khapra, Saurabh Sohoney, and Pushpak Bhattacharyya. 2010. CFILT: Resource conscious approaches for all-words domain specific WSD. In *Proceedings of the 5th International Workshop on Semantic Evaluation (Semeval-2010)*, pages 421–426, Stroudsburg, PA, USA.

Mirella Lapata and Frank Keller. 2007. An information retrieval approach to sense ranking. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL 2007, pages 348–355, Rochester, USA.

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

Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8:359–373.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, ACL 2004, pages 280–287, Barcelona, Spain.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590.

Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. 2009. Mining meaning from Wikipedia. *Int. J. Hum.-Comput. Stud.*, 67(9):716–754.

Rada Mihalcea. 2007. Using Wikipedia for automatic Word Sense Disambiguation. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, HLT-NAACL, pages 196–203, Rochester, N.Y.

George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.

Saif Mohammad and Graeme Hirst. 2006. Determining word sense dominance using a thesaurus. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2006, pages 121–128, Trento, Italy.

Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL 2010, pages 1318–1327, Uppsala, Sweden.

Roberto Navigli, Stefano Faralli, Aitor Soroa, Oier de Lacalle, and Eneko Agirre. 2011. Two birds with one stone: Learning semantic models for text categorization and Word Sense Disambiguation. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, CIKM 2011, pages 2317–2320, Glasgow, UK.

Roberto Navigli. 2005. Semi-automatic extension of large-scale linguistic knowledge bases. In *Proceedings of the 18th Internationa Florida AI Research Symposium Conference (FLAIRS),* 15–17 May 2005, pages 548–553, Clearwater Beach, Florida.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *Proceedings of the 21st National Conference on Artificial intelligence (AAAI 2006)*, pages 1400–1405, Boston, MA.

Antonio Sanfilippo, Stephen Tratz, and Michelle Gregory. 2006. Word domain disambiguation via word sense disambiguation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL 2006, pages 141–144, New York, USA.

Carlo Strapparava, Alfio Gliozzo, and Claudio Giuliano. 2004. Pattern abstraction and term similarity for Word Sense Disambiguation: IRST at Senseval-3. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*, pages 229–234, Barcelona, Spain.