Two Knowledge-based Methods for High-Performance Sense Distribution Learning

Tommaso Pasini and Roberto Navigli
{pasini,navigli}@di.uniroma1.it

Abstract

Knowing the correct distribution of senses within a corpus can potentially boost the performance of Word Sense Disambiguation (WSD) systems by many points. We present two fully automatic and language-independent methods for computing the distribution of senses given a raw corpus of sentences. Intrinsic and extrinsic evaluations show that our methods outperform the current state of the art in sense distribution learning and the strongest baselines for the most frequent sense in multiple languages and on domain-specific test sets. Our sense distributions are available at `http://trainomatic.org`.

Introduction

Word sense disambiguation (WSD) is the task of assigning the correct meaning to a word in a context, by choosing among the senses available in an inventory such as WordNet (Fellbaum 1998). WSD has received considerable interest from both academia and industry, thanks to its potential in several fields of AI, such as text understanding, machine translation and machine reading. Two main approaches have been explored to address the task, namely a supervised and a knowledge-based approach. The former exploits machine learning – e.g., SVM (Zhong and Ng 2010) and, more recently, Neural Networks (Raganato, Delli Bovi, and Navigli 2017) – in order to predict the most suitable sense of a target word occurring in a given context. The latter, instead, relies on the connection between concepts in a semantic network and aims to find the correct sense of the target word in a context by exploiting the information available in a knowledge resource, often by applying graph techniques (e.g., PageRank (Brin and Page 1998), densest subgraph approximation (Moro, Raganato, and Navigli 2014), etc.).

Compared to tasks such as part-of-speech tagging where the number of classes is limited, WSD faces the issue of a different set of meanings for each word of interest, which multiplies the size of the training set, and increases the fine granularity of sense inventories. This inherently affects the performance of the two above-mentioned approaches: supervised algorithms suffer from the lack of enough training examples, while knowledge-based systems are hampered by poor contexts – as they do not exploit local features – and the lack of connectivity for lower frequency senses (Pilehvar and Navigli 2014).

In order to cope with the so-called knowledge acquisition bottleneck and provide reliable answers for all words in the lexicon, both supervised and knowledge-based approaches resort to the most frequent sense (MFS) baseline, that is, they exploit the mode of the prior probability distribution on senses of each word. This is computed based on the sense frequencies within the largest manually annotated corpus available for the English language, i.e., SemCor (Miller et al. 1993). The MFS is normally used as a backoff strategy when the WSD system is less confident or does not have any training example for the target word at hand.

The MFS strategy has become a very strong baseline for English WSD and it is often hard to beat (Navigli, Jurgens, and Vannella 2013; Moro and Navigli 2015), due to the skewed distribution of word senses. While this is especially true for knowledge-based systems, it also holds for supervised systems whose bias towards frequent senses strictly depends on the number of examples in the training set.

Determining sense probability distributions based on SemCor comes, however, with important limitations: first, due to its size, the corpus lacks coverage of a significant part of English vocabulary (for instance, words such as *viral*, or *online*) and of many senses of ambiguous words (for instance, *cloud* in the computing sense, or *bank* as a supply held in reserve). Furthermore, it does not provide a reliable estimate of those words which, instead, are to be found in it (for instance, the corpus contains two occurrences of *tiger*, one for the animal and one for an audacious person, which does not reflect the expected probability); second, SemCor is almost 30 years old, with an outdated distribution of word uses (for example, the predominant sense of *pipe* is *tobacco pipe* while in modern English it is *tube carrying water* (McCarthy et al. 2004a)); third, there are no corpora of the same size in other languages, so sense distribution estimation from sense-tagged corpora is mostly limited to English, not to mention domain-specific estimates of word sense priors (Faralli and Navigli 2012).

Given the above limitations, we advocate that the ability to automatically learn the distribution of a word's senses is a necessary step to improve the performance of current state-of-the-art WSD systems, to enhance domain-specific WSD

and to enable multilingual WSD for supervised systems. In the last few years, various methods for automatically learning sense distributions have been proposed (McCarthy et al. 2004b; Lau et al. 2014; Bennett et al. 2016) and proven to learn better distributions, in terms of Jensen-Shannon divergence, than those extracted from a manually-annotated corpus such as SemCor. However, not many extrinsic evaluations have been conducted to prove that the learned distributions improve the disambiguation quality and offer a real alternative to those that are manually created and, even more importantly, to the best of our knowledge, no experiment has been carried out on languages other than English.

In this paper we propose two language-independent approaches to automatically learning the distribution of senses from a given corpus. Both approaches are shown to outperform the current state of the art in intrinsic and extrinsic evaluations, including domain-biased settings, while performing at least as well as the sense distributions built from SemCor.

Two Methods for Learning Sense Distributions

We present EnDi and DaD: two language-independent and fully automatic methods for sense distribution learning from raw text. Both methods share a procedure for producing a sense probability distribution for a given word at the sentence level: this procedure takes as input a lexicon $\mathcal{L}$, a raw corpus of sentences $\mathcal{C}$ and a semantic network $\mathcal{G} = (V, E)$. We assume a WordNet-like structure for $\mathcal{G}$ (Fellbaum 1998), i.e. the vertices in V are synsets that contain different lexicalizations (lemmas) of the same concept. Sentence-level sense distribution learning is performed in two steps:

- **Semantic vector computation:** in this step we compute a vector for each synset in the semantic network. Its components are all the synsets in the graph and their values can be interpreted as a measure of relatedness between the starting synset and the corresponding component.
- **Sentence-level word sense distribution:** in this step, for each sentence in $\mathcal{C}$ and for each word $w \in \mathcal{L}$ we compute a probability distribution over its senses by exploiting the lexical vectors computed in the previous step.

Semantic vector computation

The first step aims at computing a semantic vector for each synset, i.e. node, in the semantic graph that has as components all the others nodes in the graph. This probability value is computed by applying Personalized PageRank (PPR), a variant of PageRank (Brin and Page 1998) in which the uniform restart probability is changed to a custom probability. In our case, we concentrate all the restart probability mass onto the synset for which the vector is calculated, so as to increase the probability of reaching nodes in the surroundings of the synset of interest.

Running PPR with restart on a given synset s produces a semantic vector which represents the probability distribution over the synsets in the network (including s itself) of being reachable from, and thus related to, s.

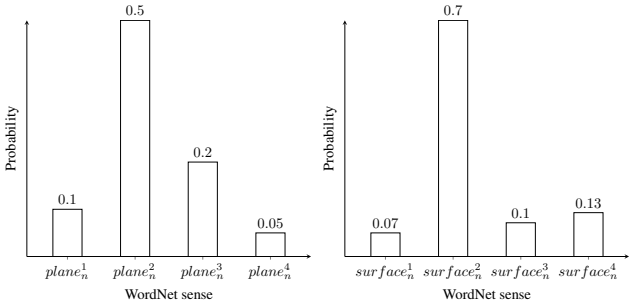

Figure 1: Probability of the senses of *plane* (left) and *surface* (right).

Shallow sense distribution learning

In the second step, each sentence in $\mathcal{C}$ in processed separately by considering all its content words[1] and building, for each of them, a probability distribution over their senses. Thus, given a word $w \in \mathcal{L}$ contained in a sentence $\sigma \in \mathcal{C}$, we want to score each of the meanings of w (drawn from the WordNet-like semantic network) with the probability of seeing that sense in the given sentence. Such probability is computed with the following formula:

$$P(s|\sigma, w) = \frac{P(\sigma|s, w)P(s|w)}{P(\sigma|w)} \quad (1)$$

$$= \frac{P(w_1, \ldots, w_n|s, w)P(s|w)}{P(w_1, \ldots, w_n|w)} \quad (2)$$

$$\propto P(w_1, \ldots, w_n|s, w)P(s|w) \quad (3)$$

$$\approx \prod_{w' \in \sigma} P(w'|s, w) \quad (4)$$

$$= \prod_{w' \in \sigma} find(\overrightarrow{PPR}_s, w') \quad (5)$$

where s is a sense of w and w is contained in σ. We approximate the probability in Equation 3 by making the independence assumption between words in the sentence and calculate the probability in Equation 4 with the function *find* in Equation 5 which returns the highest-probability synset in the first-argument vector v which has word w as one of its lexicalizations:

$$find(\overrightarrow{v}, w) = \max_{s \in C_w^v} \overrightarrow{v}(s) \quad (6)$$

where C_w^v is the set of all the components of $\overrightarrow{v}$ that have w among their lexicalizations. Once we have applied this procedure to all the sentences in the corpus $\mathcal{C}$ for each given word $w \in \mathcal{L}$, we obtain a sense probability distribution for all the sentences w occurs in. For example, given a sentence σ in the corpus (e.g. *The coordinate plane is a two-dimension surface*), a lexicon $\mathcal{L} = \{plane, surface\}$ and $\mathcal{G} =$ WordNet, the above procedure outputs two distributions (for *plane* and *surface*) as shown in Figure 1. Such distributions are used in our two methods, described below, to compute a unified distribution of senses for each word in the lexicon.

[1] We filter out non-content words and stopwords.

Sentence	plane$_n^1$ (aircraft)	plane$_n^2$ (geometry)	plane$_n^5$ (carpentry)
Two people on the **plane** died.	0.92	0.01	0.07
The flight was delayed due to trouble with the **plane**.	0.82	0.07	0.11
Only one **plane** landed successfully.	0.73	0.10	0.17
The cabinetmaker used a **plane** for the finish work.	0.20	0.18	0.62
A catalog of special **plane** curves.	0.10	0.85	0.05
$\mathcal{D}_{plane}$	0.55	0.24	0.21

Table 1: A sense distribution computation example for the word *plane*.

1) Entropy-Based Distribution Learning (EnDi)

We now introduce the first method for calculating a sense distribution for a given word. This method takes as input the lexicon $\mathcal{L}$, the set of sense distributions $\Gamma_w = \{\gamma_w^\sigma : w \in \sigma\}$ for each word $w \in \mathcal{L}$, which have been computed in the previous step, and a threshold θ.

In order to build a single sense distribution $\mathcal{D}_w$ for each word w in $\mathcal{L}$, we first select the set of sense distributions for all its sentences which have low entropy as follows:

$$\hat{\Gamma}_w = \{\gamma_w^\sigma \in \Gamma_w \,:\, \mathcal{H}(\gamma_w^\sigma) \leq \theta\} \tag{7}$$

where γ_w^σ is the distribution over w's senses in the sentence σ and $\mathcal{H}(\gamma)$ is the entropy of the input distribution γ:

$$\mathcal{H}(\gamma) = -\sum_{s \in \gamma} \gamma(s) log_2(\gamma(s))$$

As a result $\hat{\Gamma}_w$ contains only skewed sense distributions computed from sentences for which the sense bias is stronger and, therefore, the final decision is clearer. Finally the unified probability mass function $\mathcal{D}_w$ for a word w is computed so as to have, for each sense s of w, the following value:

$$\mathcal{D}_w(s) = \frac{1}{|\hat{\Gamma}_w|} \sum_{\gamma_w^\sigma \in \hat{\Gamma}_w} \gamma_w^\sigma(s) \tag{8}$$

For example, let's consider the word *plane* and 5 sentences that contain it (see Table 1, left column): we compute its sense distribution by summing the probability of each sense across the sentences and then renormalizing the results by their sum (last row of the Table).

2) Domain-Aware Distribution Learning (DaD)

The second method for sense distribution learning again takes as input the lexicon $\mathcal{L}$, the set of sense distributions $\Gamma_w = \{\gamma_w^\sigma : w \in \sigma\}$ for each word $w \in \mathcal{L}$, and a semantic network $\mathcal{G} = (V, E)$. The idea here is to exploit associations between synsets in V and domains from a fixed set $\mathcal{D}$.[2]

Note that each synset might be associated with zero, one or more domains, and that these associations come from an off-the-shelf resource, such as BabelNet domains (Camacho-Collados and Navigli 2017).

We learn the sense distribution in two steps:

1. **Domain distribution:** we compute the distribution of the domains in the shallow disambiguated corpus.
2. **Sense distribution computation:** we augment the semantic network with domain nodes and connect them to the synsets in the semantic network they are associated with. We then run Personalized PageRank on the augmented semantic graph and obtain a sense distribution over all the synsets in the graph.

Domain distribution. Given the set of sense distributions for all the words in $\mathcal{L}$ and all the sentences in the input corpus, we calculate the following probability for each domain $d \in \mathcal{D}$:

$$C(d) = \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \sum_{\substack{s \in \gamma : \\ d \in domains(s)}} \gamma(s) \tag{9}$$

where $domains(s)$ is the set of domains s is associated with. Thus, each sense in a distribution γ contributes to the probability of each domain it belongs to proportionally to its probability in γ. The average of these contributions provides the final probability of domain d in the input corpus.

The hunch behind this step is that if a corpus is domain-biased then synsets belonging to that domain should appear more often and with higher probabilities, therefore contributing to increasing the corresponding domain probability. A second hunch is that, even though the shallow distributions inherently come with some unavoidable noise, sometimes due to fine-grained sense distinctions, abstracting synsets with domains enables a coarser, hopefully more accurate and wider coverage, level.

Sense distribution computation. Now that we have a distribution over domains, we add a node for each domain to the original semantic network. We then connect with a direct edge each domain node to all the synsets it is associated with. We finally compute a probability for each synset by applying Personalized PageRank by distributing the restart probability on the domain nodes according to the domain probability distribution computed with Equation 9. Given the PageRank formula:

$$v^{(t+1)} = (1 - \alpha)v^{(0)} + \alpha M v^{(t)} \tag{10}$$

we therefore set $v^{(0)}$ to 0 except for those components (i.e. nodes) corresponding to each domain d, which are set to the corresponding domain probability $C(d)$. Thus, using the analogy of the random walker, it means that every time the

[2]The list of domains can be found at: `http://babelnet.org/javadoc/it/uniroma1/lcl/babelnet/data/BabelDomain.html`

walker decides to restart its walk, it will move to a new domain node with a certain probability.

As a result of the PageRank computation we have a distribution over all the semantic network's nodes, i.e., synsets. Note that at this stage the distribution is not specific to a given word, but is general for all synsets in $\mathcal{G}$. Thus the sense distribution of a word can be retrieved by considering the probabilities of all the synsets of that word (i.e., its senses) in the resulting PageRank vector and normalizing them so as to obtain a sense distribution of the word's senses.

Experimental Setup

We carried out both intrinsic and extrinsic evaluations in order to have a measure of how well the two methods perform both theoretically and in practice. Both methods for sense distribution learning have some parameters, namely: the semantic network, the corpus and, for the first method, the entropy threshold θ.

Semantic Network. We started from BabelNet, which is currently the largest multilingual semantic network, with around 14 million synsets covering hundreds of languages and dozens of domains (Navigli and Ponzetto 2012). BabelNet is a superset of WordNet, Wikipedia, Wiktionary and other resources and therefore is richer in terms of lexicalizations and semantic relations than any of the resources it integrates. However, because – similarly to alternatives in the literature – we focused only on common nouns, we followed Pasini and Navigli (2017) and chose the WordNet-induced subgraph of BabelNet as the underlying network for the semantic vector computation. In other words, the graph contained only WordNet synset nodes, but with the considerably larger set of relation edges and multilingual lexicalizations coming from BabelNet.

Corpus. We chose Wikipedia as our input corpus because it is available in hundreds of languages and it covers all domains of human knowledge. We used the October 2014 dump of Wikipedia.

Entropy threshold. For EnDi we tried different values of the threshold θ, ranging from 0.1 to 4.0 with step 0.1, and tested the results on an in-house development set of 25 lemmas for which we computed the sense distribution in Wikipedia and then selected the value of θ based on the best results in terms of similarity to the SemCor distribution (as explained below). We thus set θ to 1.0.

Comparison sense distributions and gold standard. We compared the two sense distribution learning methods against several alternative methods for deriving sense distributions for words, namely:

- **EuroSense** (Delli Bovi et al. 2017): a sense-annotated resource based on the multilingual joint disambiguation of the Europarl corpus (Koehn 2005). For a given word, its sense distribution is obtained by computing the normalized frequency of its senses in the corpus.
- **LexSemTM** (Bennett et al. 2016): an approach which builds on top of (Lau et al. 2014) and exploits sense glosses and usage examples of the target lemma to build a topic model and then a distribution of the target word's senses.
- **WordNet and BabelNet degree:** we created sense distributions based on the normalized out-degrees of the various senses of the target word in two different semantic networks: WordNet and BabelNet. Given one of the two graphs, we calculated the distribution as follows:

$$\mathcal{D}_w(s) = \frac{\text{out-deg}(s)}{\sum_{s' \in senses(w)} \text{out-deg}(s')}$$

- **SemCor gold-standard sense distribution:** we also compared against the sense distribution computed based on sense frequencies in SemCor (Miller et al. 1993). Notice that this is a gold-standard distribution, as it is the only distribution obtained from manually-annotated data.

Intrinsic Evaluation

Evaluation measures

For the intrinsic evaluation we evaluated the similarity between the two distributions learned with our entropy and domain-based methods and all other comparison sense distributions introduced above. We used two different measures for performing the comparison, i.e., Jensen-Shannon divergence and Weighted Overlap. Both measures were computed separately for each pair of distributions and then averaged by the total number of words.

Jensen-Shannon divergence (JSD). This measure is based on the Kullback-Leibler divergence and equals 0 when the two distributions are identical, and is greater than 0 when they are different in some way. It is computed as follows:

$$JSD(\gamma, \gamma') = \frac{D(\gamma, M)}{2} + \frac{D(\gamma', M)}{2} \tag{11}$$

where $M = \frac{\gamma+\gamma'}{2}$ and D is the Kullback-Leibler divergence which is given by the following formula:

$$D(\gamma, \gamma') = \sum_s \gamma(s) log(\frac{\gamma(s)}{\gamma'(s)}) \tag{12}$$

where, in our case, s are synsets in our sense distributions.

Weighted Overlap. This measure (Pilehvar, Jurgens, and Navigli 2013, WO) determines how similar the sense rankings of the two distributions are. It is 1 when the two distributions have the same ranking of the components and lower than 1 when they are different. It is defined as follows:

$$WO(\gamma, \gamma') = \sum_{i=1}^{|O|} \frac{(r_i + r'_i)^{-1}}{(2i)^{-1}} \tag{13}$$

where O is the intersection of the components of γ and γ' and r_i and r'_i are the ranks of the *i-th* component in the

Method	JSD_{gold}	WO_{gold}	JSD_{sys}	WO_{sys}
EnDi	0.29	0.70	**0.06**	0.89
DaD	0.17	**0.91**	0.12	**0.92**
LexSemTM	0.29	0.67	0.07	0.89
EuroSense	0.60	0.39	0.24	0.75
BabelNet Degree	0.09	0.87	0.09	0.87
WordNet Degree	**0.07**	0.88	0.07	0.88

Table 2: Similarity with SemCor in terms of Jensen-Shannon divergence and Weighted Overlap (*gold* evaluates against all words in SemCor; *sys* evaluates only against the words for which each method can provide a sense distribution).

respective distribution γ and γ'. The rank of a component (i.e., sense) of the distribution vector is the position at which the component can be found in the distribution vector when sorted in descending order. The Weighted Overlap is thus a measure that does not consider the value of the components in the distributions, but only their ranking.

These two measures provide different insights about how the sense frequencies of a given word are distributed both numerically and when we only consider the components position when the distributions are sorted by value.

Results

Similarity to SemCor distributions. The first experiment we performed aimed at investigating the similarity between the various automatically learned distributions (both with our two methods and the comparison distributions) and the gold-standard SemCor distribution. In Table 2 we show the JSD and WO (note that for JSD the lower the better, while for WO the higher the better) averaged among all the words in the test set. Both measures were computed, first, by considering all the lemmas in the test set and assigning 1 and 0 to JSD and WO, respectively, when the method was not able to build a distribution for a given lemma (second and third column), and then considering only the lemmas for which the method was able to build the distribution (fourth and fifth columns of the table). As can be seen both our methods built distributions that are generally most similar to SemCor, in terms of both JSD and WO, than the state-of-the-art LexSemTM and that are either better or on a par with alternative approaches.

More in detail, DaD performs best in the gold setting, showing wide coverage of words, but a bit worse in numerical terms according to JSD_{sys}. In contrast, EnDi performs best in terms of JSD_{sys}, due to its ability to prune out noisy sentences, slightly worse in the ranking evaluation and on a par with LexSemTM across the board. Degrees fare well, especially on JSD, but, as we will see, their extrinsic evaluation results turn out to be considerably lower.

We show lemma coverage in Table 3: DaD and the degree-based distributions have the highest coverage of words, which is WordNet's, while EnDi and LexSemTM – due to filtering mechanisms – and EuroSense – due to lack of sense annotations – have much lower word coverage.

Method	Missing Lemmas
EnDi	2655
DaD	23
LexSemTM	2783
EuroSense	5378
BabelNet Degree	23
WordNet Degree	23

Table 3: Lemmas for which a method was not able to build a distribution.

Method	JSD	WO
EnDi	**0.099**†	**0.937**
DaD	0.204	0.902
LexSemTM	0.116†	0.932
EuroSense	0.344	0.713
BabelNet Degree	0.224	0.832
WordNet Degree	0.166	0.858
SemCor	0.255	0.837

Table 4: Similarity with the gold standard from Bennett et al. (2016) in terms of JSD and Weighted Overlap. Values tagged with † are statistical significant for $p < 0.1$.

Similarity to Bennett et al.'s (2016) distributions. So far we have shown that our methods produced high-quality sense distributions when compared against SemCor. While this is a good result, we should consider that SemCor dates back to almost 30 years ago and since then sense distributions have surely changed over time for a number of ambiguous words (e.g. *troll*, *tweet*, etc.). To work on more recent data, we performed a second intrinsic evaluation using a gold standard dataset proposed by Bennett et al. (2016), which provides distributions manually annotated for 50 lemmas. In this experiment we also evaluated the SemCor-derived distribution against the 50-lemma gold standard. In Table 4 we report the results in terms of JSD and WO on this dataset[3]: our methods have lower JSD values than SemCor distribution. Another interesting result is that both WordNet and BabelNet degree baselines also beat SemCor by 0.09 and 0.03 points, while EuroSense achieved the worst results. LexSemTM, instead, scored pretty well according to both measures, achieving 0.116 on JSD and 0.932 on WO. DaD on the other hand scored better than SemCor but worse than LexSemTM on JSD and slightly worse on WO; in contrast, EnDi turned out to be the best method according to the JSD measure and was equivalent to LexSemTM on WO, achieving the state of the art on this dataset. Note also that JSD values are statistical significant for $p < 0.1$.

Extrinsic Evaluation

We now move to the extrinsic evaluation, which was performed in the context of all-words Word Sense Disambiguation. It is well known in the literature that always outputting the most frequent sense for each ambiguous word in context

[3]We note that, here, all the systems were able to generate a distribution for each lemma.

Method	Precision	Recall	F1
EnDi	0.66	0.66	0.66
DaD	0.61	0.61	0.61
LexSemTM	0.51	0.48	0.49
BabelNet degree	0.51	0.38	0.43
WordNet degree	0.55	0.44	0.49
WordNet MFS	0.68	0.68	**0.68**

Table 5: Most Frequent Sense performance on all Senseval/SemEval test sets from Raganato et al. (2017).

– the so-called Most Frequent Sense (MFS) baseline – is a hard-to-beat disambiguation strategy (Navigli 2009). The MFS for English is usually calculated based on frequencies as reported in WordNet, which exploit those in the SemCor corpus. Therefore, we can evaluate each sense distribution method by (1) for each word, identifying the predominant (i.e., highest-probability) sense according to the returned sense distribution, and (2) always outputting that sense every time in a WSD dataset we are required to disambiguate the given word. By applying this procedure, we compared the results of the various approaches against the WordNet MFS and BabelNet and WordNet degree. As test sets, we used the benchmark from Raganato, Camacho-Collados, and Navigli (2017) which is the union of all the past Senseval and SemEval for all-words WSD, namely: Senseval-2 (Edmonds and Cotton 2001), Senseval-3 (Snyder and Palmer 2004), SemEval-2007 (Navigli, Litkowski, and Hargraves 2007), SemEval-2013 (Moro and Navigli 2015) and SemEval-2015 (Moro and Navigli 2015).

As shown in Table 5, not only do both EnDi and DaD beat LexSemTM by several points, but they also, especially EnDi, have performance close to the WordNet MFS baseline with a gap of 2 and 7 points, while both BabelNet and WordNet scored lower or equal to LexSemTM. This result corroborates the consistently good results of EnDi in the intrinsic evaluation. Moreover we think that this is a very significant outcome since a fully automatic system was able to learn sense distributions that perform very close to a hard-to-beat baseline obtained from a manually sense-annotated dataset.

This is again a good result, but, as mentioned above, annotated data exists for English from which usable sense distributions can be derived (however, this data is outdated and limited in size, while our distributions can be updated over time and cover much of the lexicon). To further show the effectiveness of our methods, we performed experiments in other languages, for which manually annotated data is not available on a reasonable scale. Our methods are indeed language-independent and, thanks to BabelNet lexicalizations, we can apply them to arbitrary languages. We therefore calculated sense distributions from the Italian and Spanish Wikipedia (dumps from Oct. 2014) and compared their performance on the SemEval-2015 all-words multilingual Word Sense Disambiguation task. We set the threshold θ for Italian and Spanish to 1.0 and 0.001 experimentally, equally to how we did for English. The difference is due to the fact that the Spanish part of BabelNet contains more ambiguous data.

Method	Precision	Recall	F1
EnDi	0.60	0.50	0.55
DaD	0.67	0.56	**0.61**
BabelNet Degree	0.57	0.52	0.54
BFS	0.54	0.50	0.52

Table 6: Comparison of Most Frequent Sense performance for Italian on the SemEval-2015 WSD task.

Method	Precision	Recall	F1
EnDi	0.58	0.48	0.52
DaD	0.64	0.54	**0.58**
BabelNet Degree	0.56	0.52	0.54
BFS	0.55	0.51	0.53

Table 7: Comparison of the Most Frequent Sense performance for Spanish on the SemEval-2015 WSD task.

Results for Italian are shown in Table 6. We compared our two methods against the BabelNet Degree and the BabelNet First Sense (BFS) baseline, a dictionary-based baseline which was used as baseline in the task (due to the lack of a manually annotated dataset from which an MFS could be estimated in Italian). The results show that our method performs better than the current best baseline. In particular, DaD outperforms the BFS by 9 F1 points. The gap is even bigger when looking at precision, where our two methods gain from 6 to 13 points, while recall is increased by 6 points with DaD. BabelNet degree also performed better than the BFS but anyway less effectively than both our systems. A similar trend is observed for Spanish (Table 7), with DaD attaining a 5% F1 improvement over the Spanish BFS.

Domain-Specific Evaluation

We next investigated how well our methods were able to produce skewed distributions of senses in specific domains.

Learning domain-specific sense distributions. To bias sense distributions towards specific domains, we exploited the 34 domains available in BabelNet (Camacho-Collados and Navigli 2017). For each domain d, we collected all the synsets in BabelNet tagged with that domain and which contain a Wikipedia page. We then used all the sentences from the retrieved pages to build a corpus for domain d. On each corpus, we then applied EnDi and DaD to obtain sense distributions that were biased towards the domain of interest.

Evaluation. We tested our domain-biased sense distributions against domain-specific documents from the same SemEval-13 and SemEval-15 test sets used in the above extrinsic experiment, as tagged by the task organizers. In contrast to the above experiments, results are therefore reported individually for each domain where EnDi and DaD used the corresponding distributions learned for that domain[4].

[4]We note that, here, the only additional information provided as input to the methods was the domain label.

Method	Metrics	Biology	Climate	Finance	Medicine	Politics	Social Issues	Sport
EnDi	F1	0.71	0.53	0.60	0.46	0.62	0.63	**0.57**
DaD	F1	**0.79**	**0.63**	**0.64**	**0.64**	**0.67**	**0.68**	0.54
LexSemTM	F1	0.56	0.47	0.49	0.42	0.51	0.52	0.34
WN MFS	F1	0.61	0.59	0.52	0.50	0.64	0.58	0.56

Table 8: Domain evaluation on SemEval-2013 WSD.

Method	Domain	Precision	Recall	F1
EnDi	Math & Computer	0.65	0.61	0.63
	Biomedicine	0.65	0.62	0.63
DaD	Math & Computer	0.66	0.66	0.66
	Biomedicine	0.63	0.63	0.63
LexSemTM	Math & Computer	0.48	0.47	0.47
	Biomedicine	0.64	0.61	0.63
WN MFS	Math & Computer	0.48	0.46	0.47
	Biomedicine	0.70	0.67	**0.68**

Table 9: Domain evaluation on SemEval-2015 WSD.

Results are shown in Tables 8 and 9. EnDi outperforms LexSemTM on all domains across the two datasets. While the latter performances are always lower than the WordNet MFS baseline, EnDi is instead able to surpass the baseline on 4 out of 7 domains on SemEval-2013 and on one of the two domains of SemEval-2015. As regards DaD, not only does it consistently beat LexSemTM, achieving up to 23 points higher in F1, but it also beats the WordNet MFS on every domain but one of SemEval-2013 and one of the two domains of SemEval-2015, performing on average 8 F1 points higher.

Related Work

Word Sense Disambiguation (WSD) has long been studied within the AI and the NLP communities, which have come up with various approaches to the problem (Navigli 2009). A mainstream direction has been supervised WSD, which trains a machine learning classifier with large amounts of training data. The most successful approaches have been based on SVM (Zhong and Ng 2010) and, more recently, on neural LSTM architectures (Yuan et al. 2016; Raganato, Delli Bovi, and Navigli 2017). A second popular approach has been knowledge-based WSD, which exploits knowledge resources like WordNet and BabelNet to perform the task. Examples include Babelfy (Moro, Raganato, and Navigli 2014) and PPR (Agirre, de Lacalle, and Soroa 2014), both performing random walks to identify the most relevant meanings for words in context. Both directions have their own limitations, but one they have in common is the well-known knowledge acquisition bottleneck. In fact, supervised systems cannot perform well in the absence of large amounts of training data, a requirement that is even amplified with neural network classifiers. On the other hand, knowledge-based systems need well structured, rich semantic networks. To mitigate this problem, systems started to use estimates of the most frequent sense (MFS) of a word as a backoff strategy for those words for which not enough information was available. The MFS has proven a hard-to-beat baseline (McCarthy et al. 2007) and it is able to improve a system's performance by many points (Pasini and Navigli 2017). Several approaches have been proposed which learn more precise sense distributions in order to improve MFS performance. A seminal work was by McCarthy et al. (2004b), who relied on distributionally similar words in order to find the predominant meaning of the target word. Subsequent studies, instead, put the focus on distribution learning within domains (Chan and Ng 2006). More recently, Bhingardive et al. (2015) exploited word embeddings to identify the most frequent sense by comparing the word vector against all the vectors of its senses.

A different approach was taken by Jin et al. (2009), who analyzed the entropy of the distributions of senses in order to decide when it is better to use the MFS as output and when it is not. The most recent effort in this direction, and the previous state of the art, is the work by Bennett et al. (2016), which in its turn was based on (Lau et al. 2014) and adopted topic modeling and word sense induction techniques (Lau et al. 2012) to learn word sense distributions. With EnDi and DaD, instead, we exploit Personalized PageRank on a WordNet-like semantic network, and, respectively, the entropy of a sense distribution and domain profiling to learn the probability distribution of senses in a given corpus, achieving state-of-the-art performance.

Conclusion

In this paper we presented EnDi and DaD, two knowledge-based, language-independent methods for learning the distributions of senses from an input corpus without relying on manual training data. They have been shown to perform well on intrinsic and extrinsic evaluations, outperforming the other baselines. Thanks to effective entropy-based filtering, EnDi outperforms LexSemTM, the previous state of the art in sense distribution learning, in all evaluations for the English language. We also showed that, in contrast to other approaches, both methods scale well to other languages, with DaD surpassing all alternative methods in the two SemEval multilingual disambiguation tasks. Thanks to its domain awareness, not only has DaD proven to generalize well across languages, but also to surpass the F1 performance of the hard-to-beat WordNet MFS on 8 out of 9 domains from two SemEval tasks. LexSemTM is, instead, surpassed by both methods across all domains.

Our data is available at `http://trainomatic.org`. As future work, we plan to use our sense distributions as a backoff strategy to increase the performance of disambiguation systems, especially in those tasks which lack manually annotated training data.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487.

References

Agirre, E.; de Lacalle, O. L.; and Soroa, A. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* 40(1):57–84.

Bennett, A.; Baldwin, T.; Lau, J. H.; McCarthy, D.; and Bond, F. 2016. LexSemTm: A semantic dataset based on all-words unsupervised sense distribution learning. In *Proc. of ACL*, 1513 – 1524.

Bhingardive, S.; Singh, D.; Rudramurthy, V.; Redkar, H.; and Bhattacharyya, P. 2015. Unsupervised most frequent sense detection using word embeddings. In *Proc. of NAACL*, 1238–1243.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* 30(1–7):107–117.

Camacho-Collados, J., and Navigli, R. 2017. BabelDomains: Large-scale domain labeling of lexical resources. In *Proc. of EACL 2017*, 223–228.

Chan, Y. S., and Ng, H. T. 2006. Estimating class priors in domain adaptation for word sense disambiguation. In *Proc. of ACL*, 89–96.

Delli Bovi, C.; Camacho-Collados, J.; Raganato, A.; and Navigli, R. 2017. EuroSense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proc. of ACL*, volume 2, 594–600.

Edmonds, P., and Cotton, S. 2001. Senseval-2: overview. In *Proc. of SensEval 2*, 1–5.

Faralli, S., and Navigli, R. 2012. A New Minimally-supervised Framework for Domain Word Sense Disambiguation. In *Proceedings of EMNLP 2012*, 1411–1422.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, Mass.: MIT Press.

Jin, P.; McCarthy, D.; Koeling, R.; and Carroll, J. 2009. Estimating and exploiting the entropy of sense distributions. In *Proc. of NAACL*, 233–236.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*.

Lau, J. H.; Cook, P.; McCarthy, D.; Newman, D.; and Baldwin, T. 2012. Word sense induction for novel sense detection. In *EACL*, 591–601.

Lau, J. H.; Cook, P.; McCarthy, D.; Gella, S.; and Baldwin, T. 2014. Learning word sense distributions, detecting unattested senses and identifying novel senses using topic models. In *Proc. of ACL*, 259–270.

McCarthy, D.; Koeling, R.; ; and Weeds, J. 2004a. Ranking wordnet senses automatically. Technical Report 569, Department of Informatics, University of Sussex.

McCarthy, D.; Koeling, R.; Weeds, J.; and Carroll, J. 2004b. Finding predominant senses in untagged text. In *Proc. of ACL 2004*, 280–287.

McCarthy, D.; Koeling, R.; Weeds, J.; and Carroll, J. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics* 33(4):553–590.

Miller, G. A.; Leacock, C.; Tengi, R.; and Bunker, R. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, 303–308.

Moro, A., and Navigli, R. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proc. of SemEval-2015*.

Moro, A.; Raganato, A.; and Navigli, R. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)* 2:231–244.

Navigli, R., and Ponzetto, S. P. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193:217–250.

Navigli, R.; Jurgens, D.; and Vannella, D. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proc. of SemEval 2013*, 222–231.

Navigli, R.; Litkowski, K. C.; and Hargraves, O. 2007. Semeval-2007 task 07: Coarse-grained English all-words task. In *Proc. of SemEval-2007*, 30–35.

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

Pasini, T., and Navigli, R. 2017. Train-o-matic: Large-scale supervised word sense disambiguation inmultiple languages without manual training data. In *Proc. of EMNLP 2017*, 78–88.

Pilehvar, M. T., and Navigli, R. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics* 40(4):837–881.

Pilehvar, M. T.; Jurgens, D.; and Navigli, R. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proc. of ACL*, 1341–1351.

Raganato, A.; Camacho-Collados, J.; and Navigli, R. 2017. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proc. of EACL*, 99–110.

Raganato, A.; Delli Bovi, C.; and Navigli, R. 2017. Neural sequence learning models for word sense disambiguation. In *Proc. of EMNLP*, 1167–1178.

Snyder, B., and Palmer, M. 2004. The english all-words task. In *Proc. of Senseval-3*, 41–43.

Yuan, D.; Richardson, J.; Doherty, R.; Evans, C.; and Altendorf, E. 2016. Semi-supervised word sense disambiguation with neural models. *Proc. of COLING* 1374–1385.

Zhong, Z., and Ng, H. T. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proc. of ACL*, 78–83.