Unsupervised Does Not Mean Uninterpretable: The Case for Word Sense Induction and Disambiguation

Alexander Panchenko[‡], **Eugen Ruppert**[‡], **Stefano Faralli**[†], **Simone Paolo Ponzetto**[†] **and Chris Biemann**[‡]

[‡]Language Technology Group, Computer Science Dept., University of Hamburg, Germany
[†]Web and Data Science Group, Computer Science Dept., University of Mannheim, Germany
{panchenko,ruppert,biemann}@informatik.uni-hamburg.de
{faralli,simone}@informatik.uni-mannheim.de

Abstract

The current trend in NLP is the use of highly opaque models, e.g. neural networks and word embeddings. While these models yield state-of-the-art results on a range of tasks, their drawback is poor interpretability. On the example of word sense induction and disambiguation (WSID), we show that it is possible to develop an interpretable model that matches the state-of-the-art models in accuracy. Namely, we present an unsupervised, knowledge-free WSID approach, which is interpretable at three levels: word sense inventory, sense feature representations, and disambiguation procedure. Experiments show that our model performs on par with state-of-the-art word sense embeddings and other unsupervised systems while offering the possibility to justify its decisions in human-readable form.

1 Introduction

A word sense disambiguation (WSD) system takes as input a target word t and its context C. The system returns an identifier of a word sense s_i from the word sense inventory $\{s_1, ..., s_n\}$ of t, where the senses are typically defined manually in advance. Despite significant progress in methodology during the two last decades (Ide and Véronis, 1998; Agirre and Edmonds, 2007; Moro and Navigli, 2015), WSD is still not widespread in applications (Navigli, 2009), which indicates the need for further progress. The difficulty of the problem largely stems from the lack of domain-specific training data. A fixed sense inventory, such as the one of WordNet (Miller, 1995), may contain irrelevant senses for the given application and at the same time lack relevant domain-specific senses. Word sense induction from domain-specific corpora is a supposed to solve this problem. However, most approaches to word sense induction and disambiguation, e.g. (Schütze, 1998; Li and Jurafsky, 2015; Bartunov et al., 2016), rely on clustering methods and dense vector representations that make a WSD model uninterpretable as compared to knowledge-based WSD methods.

Interpretability of a statistical model is important as it lets us understand the reasons behind its predictions (Vellido et al., 2011; Freitas, 2014; Li et al., 2016). Interpretability of WSD models (1) lets a user understand why in the given context one observed a given sense (e.g., for educational applications); (2) performs a comprehensive analysis of correct and erroneous predictions, giving rise to improved disambiguation models.

The contribution of this paper is an interpretable unsupervised knowledge-free WSD method. The novelty of our method is in (1) a technique to disambiguation that relies on induced inventories as a pivot for learning sense feature representations, (2) a technique for making induced sense representations interpretable by labeling them with hypernyms and images.

Our method tackles the interpretability issue of the prior methods; it is interpretable at the levels of (1) sense inventory, (2) sense feature representation, and (3) disambiguation procedure. In contrast to word sense induction by context clustering (Schütze (1998), inter alia), our method constructs an explicit word sense inventory. The method yields performance comparable to the state-of-the-art unsupervised systems, including two methods based on word sense embeddings. An open source implementation of the method featuring a live demo of several pre-trained models is available online.[1]

[1]http://www.jobimtext.org/wsd

2 Related Work

Multiple designs of WSD systems were proposed (Agirre and Edmonds, 2007; Navigli, 2009). They vary according to the level of supervision and the amount of external knowledge used. Most current systems either make use of lexical resources and/or rely on an explicitly annotated sense corpus.

Supervised approaches use a sense-labeled corpus to train a model, usually building one sub-model per target word (Ng, 1997; Lee and Ng, 2002; Klein et al., 2002; Wee, 2010). The IMS system by Zhong and Ng (2010) provides an implementation of the supervised approach to WSD that yields state-of-the-art results. While supervised approaches demonstrate top performance in competitions, they require large amounts of sense-labeled examples per target word.

Knowledge-based approaches rely on a lexical resource that provides a sense inventory and features for disambiguation and vary from the classical Lesk (1986) algorithm that uses word definitions to the Babelfy (Moro et al., 2014) system that uses harnesses a multilingual lexical-semantic network. Classical examples of such approaches include (Banerjee and Pedersen, 2002; Pedersen et al., 2005; Miller et al., 2012). More recently, several methods were proposed to learn sense embeddings on the basis of the sense inventory of a lexical resource (Chen et al., 2014; Rothe and Schütze, 2015; Camacho-Collados et al., 2015; Iacobacci et al., 2015; Nieto Piña and Johansson, 2016).

Unsupervised knowledge-free approaches use neither handcrafted lexical resources nor hand-annotated sense-labeled corpora. Instead, they induce word sense inventories automatically from corpora. Unsupervised WSD methods fall into two main categories: context clustering and word ego-network clustering.

Context clustering approaches, e.g. (Pedersen and Bruce, 1997; Schütze, 1998), represent an instance usually by a vector that characterizes its context, where the definition of context can vary greatly. These vectors of each instance are then clustered. Multi-prototype extensions of the skip-gram model (Mikolov et al., 2013) that use no predefined sense inventory learn one embedding word vector per one word sense and are commonly fitted with a disambiguation mechanism (Huang et al., 2012; Tian et al., 2014; Neelakantan et al., 2014; Bartunov et al., 2016; Li and Jurafsky, 2015; Pelevina et al., 2016). Comparisons of the *AdaGram* (Bartunov et al., 2016) to (Neelakantan et al., 2014) on three SemEval word sense induction and disambiguation datasets show the advantage of the former. For this reason, we use *AdaGram* as a representative of the state-of-the-art word sense embeddings in our experiments. In addition, we compare to SenseGram, an alternative sense embedding based approach by Pelevina et al. (2016). What makes the comparison to the later method interesting is that this approach is similar to ours, but instead of sparse representations the authors rely on word embeddings, making their approach less interpretable.

Word ego-network clustering methods (Lin, 1998; Pantel and Lin, 2002; Widdows and Dorow, 2002; Biemann, 2006; Hope and Keller, 2013) cluster graphs of words semantically related to the ambiguous word. An ego network consists of a single node (ego) together with the nodes they are connected to (alters) and all the edges among those alters (Everett and Borgatti, 2005). In our case, such a network is a local neighborhood of one word. Nodes of the ego-network can be (1) words semantically similar to the target word, as in our approach, or (2) context words relevant to the target, as in the *UoS* system (Hope and Keller, 2013). Graph edges represent semantic relations between words derived using corpus-based methods (e.g. distributional semantics) or gathered from dictionaries. The sense induction process using word graphs is explored by (Widdows and Dorow, 2002; Biemann, 2006; Hope and Keller, 2013). Disambiguation of instances is performed by assigning the sense with the highest overlap between the instance's context words and the words of the sense cluster. Véronis (2004) compiles a corpus with contexts of polysemous nouns using a search engine. A word graph is built by drawing edges between co-occurring words in the gathered corpus, where edges below a certain similarity threshold were discarded. His HyperLex algorithm detects hubs of this graph, which are interpreted as word senses. Disambiguation is this experiment is performed by computing the distance between context words and hubs in this graph.

Di Marco and Navigli (2013) presents a comprehensive study of several graph-based WSI methods including Chinese Whispers, HyperLex, curvature clustering (Dorow et al., 2005). Besides, authors propose two novel algorithms: Balanced Maximum Spanning Tree Clustering and Squares (B-MST), Triangles and Diamonds (SquaT++). To construct graphs, authors use first-order and second-order relations extracted from a background corpus as well as keywords from snippets. This research goes beyond intrinsic evaluations of induced senses and measures impact of the WSI in the context of an information retrieval via clustering and diversifying Web search results. Depending on the dataset, HyperLex, B-MST or Chinese-Whispers provided the best results.

Our system combines several of above ideas and adds features ensuring interpretability. Most notably, we use a word sense inventory based on clustering word similarities (Pantel and Lin, 2002); for disambiguation we rely on syntactic context features, co-occurrences (Hope and Keller, 2013) and language models (Yuret, 2012).

Interpretable approaches. The need in methods that interpret results of opaque statistical models is widely recognised (Vellido et al., 2011; Vellido et al., 2012; Freitas, 2014; Li et al., 2016; Park et al., 2016). An interpretable WSD system is expected to provide (1) a human-readable sense inventory, (2) human-readable reasons why in a given context c a given sense s_i was detected. Lexical resources, such as WordNet, solve the first problem by providing manually-crafted definitions of senses, examples of usage, hypernyms, and synonyms. The BabelNet (Navigli and Ponzetto, 2010) integrates all these sense representations, adding to them links to external resources, such as Wikipedia, topical category labels, and images representing the sense. The unsupervised models listed above do not feature any of these representations making them much less interpretable as compared to the knowledge-based models. Ruppert et al. (2015) proposed a system for visualising sense inventories derived in an unsupervised way using graph-based distributional semantics. Panchenko (2016) proposed a method for making sense inventory of word sense embeddings interpretable by mapping it to BabelNet.

Our approach was inspired by the knowledge-based system Babelfy (Moro et al., 2014). While the inventory of Babelfy is interpretable as it relies on BabelNet, the system provides no underlying reasons behind sense predictions. Our objective was to reach interpretability level of knowledge-based models within an unsupervised framework.

3 Method: Unsupervised Interpretable Word Sense Disambiguation

Our unsupervised word sense disambiguation method consist of the five steps illustrated in Figure 1: extraction of context features (Section 3.1); computing word and feature similarities (Section 3.2); word sense induction (Section 3.3); labeling of clusters with hypernyms and images (Section 3.4), disambiguation of words in context based on the induced inventory (Section 3.5), and finally interpretation of the model (Section 3.6). Feature similarity and co-occurrence computation steps (drawn with a dashed lines) are optional, since they did not consistently improve performance.

3.1 Extraction of Context Features

The goal of this step is to extract word-feature counts from the input corpus. In particular, we extract three types of features:

Dependency Features. These feature represents a word by a syntactic dependency such as "nn(•,writing)" or "prep_at(sit,•)", extracted from the Stanford Dependencies (De Marneffe et al., 2006) obtained with the the PCFG model of the Stanford parser (Klein and Manning, 2003). Weights are computed using the Local Mutual Information (LMI) (Evert, 2005). One word is represented with 1000 most significant features.

Co-occurrence Features. This type of features represents a word by another word. We extract the list of words that significantly co-occur in a sentence with the target word in the input corpus based on the log-likelihood as word-feature weight (Dunning, 1993).

Language Model Feature. This type of features are based on a trigram model with Kneser-Ney smoothing (Kneser and Ney, 1995). In particular, a word is represented by (1) right and left context words, e.g. "office_•_and", (2) two preceding words "new_office_•", and (3) two succeeding words, e.g. "•_and_chairs". We use the conditional probabilities of the resulting trigrams as word-feature weights.

3.2 Computing Word and Feature Similarities

The goal of this step is to build a graph of word similarities, such as (table, chair, 0.78). We used the *JoBimText* framework (Biemann and Riedl,

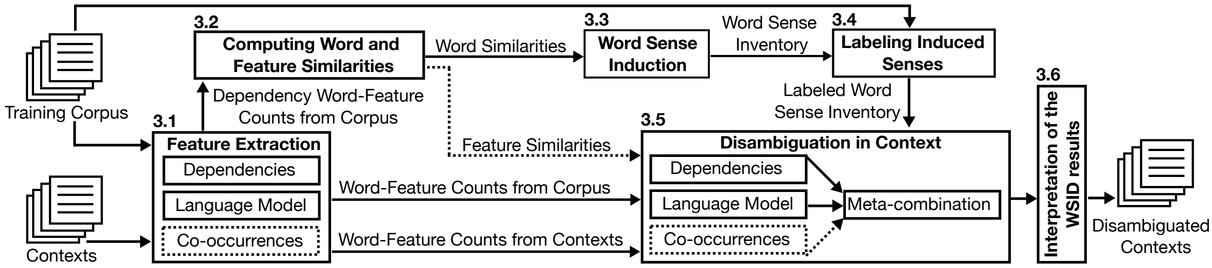

Figure 1: Outline of our unsupervised interpretable method for word sense induction and disambiguation.

2013) as it yields comparable performance on semantic similarity to state-of-the-art dense representations (Mikolov et al., 2013) compared on the WordNet as gold standard (Riedl, 2016), but is interpretable as word are represented by sparse interpretable features. Namely we use dependency-based features as, according to prior evaluations, this kind of features provides state-of-the-art semantic relatedness scores (Padó and Lapata, 2007; Van de Cruys, 2010; Panchenko and Morozova, 2012; Levy and Goldberg, 2014).

First, features of each word are ranked using the LMI metric (Evert, 2005). Second, the word representations are pruned keeping 1000 most salient features per word and 1000 most salient words per feature. The pruning reduces computational complexity and noise. Finally, word similarities are computed as a number of common features for two words. This is again followed by a pruning step in which only the 200 most similar terms are kept to every word. The resulting word similarities are browsable online.[2]

Note that while words can be characterized with distributions over features, features can vice versa be characterized by a distribution over words. We use this duality to compute feature similarities using the same mechanism and explore their use in disambiguation below.

3.3 Word Sense Induction

We induce a sense inventory by clustering of ego-network of similar words. In our case, an inventory represents senses by a word cluster, such as "chair, bed, bench, stool, sofa, desk, cabinet" for the "furniture" sense of the word "table".

The sense induction processes one word t of the distributional thesaurus T per iteration. First, we retrieve nodes of the ego-network G of t being the N most similar words of t according to T (see Figure 2 (1)). Note that the target word t itself is not part of the ego-network. Second, we connect each node in G to its n most similar words according to T. Finally, the ego-network is clustered with Chinese Whispers (Biemann, 2006), a non-parametric algorithm that discovers the number of senses automatically. The n parameter regulates the granularity of the inventory: we experiment with $n \in \{200, 100, 50\}$ and $N = 200$.

The choice of Chinese Whispers among other algorithms, such as HyperLex (Véronis, 2004) or MCL (Van Dongen, 2008), was motivated by the absence of meta-parameters and its comparable performance on the WSI task to the state-of-the-art (Di Marco and Navigli, 2013).

3.4 Labeling Induced Senses with Hypernyms and Images

Each sense cluster is automatically labeled to improve its interpretability. First, we extract hypernyms from the input corpus using Hearst (1992) patterns. Second, we rank hypernyms relevant to the cluster by a product of two scores: the *hypernym relevance* score, calculated as $\sum_{w \in cluster} sim(t, w) freq(w, h)$, and the *hypernym coverage* score, calculated as $\sum_{w \in cluster} \min(freq(w, h), 1)$. Here the $sim(t, w)$ is the relatedness of the cluster word w to the target word t, and the $freq(w, h)$ is the frequency of the hypernymy relation (w, h) as extracted via patterns. Thus, a high-ranked hypernym h has high relevance, but also is confirmed by several cluster words. This stage results in a ranked list of labels that specify the word sense, for which we here show the first one, e.g. "table (furniture)" or "table (data)".

Faralli and Navigli (2012) showed that web search engines can be used to bootstrap sense-related information. To further improve interpretability of induced senses, we assign an image to each word in the cluster (see Figure 2) by query-

[2]Select the "JoBimViz" demo and then the "Stanford (English)" model: `http://www.jobimtext.org`.

ing the Bing image search API[3] using the query composed of the target word and its hypernym, e.g. "jaguar car". The first hit of this query is selected to represent the induced word sense.

Algorithm 1: Unsupervised WSD of the word t based on the induced word sense inventory I.

```
input : Word t, context features C, sense inventory I,
        word-feature table F, use largest cluster
        back-off LCB, use feature expansion FE.
output: Sense of the target word t in inventory I and
        confidence score.
1  S ← getSenses (I, t)
2  if FE then
3  |  C ← featureExpansion(C)
4  end
5  foreach (sense, cluster) ∈ S do
6  |  α[sense] ← {}
7  |  foreach w ∈ cluster do
8  |  |  foreach c ∈ C do
9  |  |  |  α[sense] ← α[sense] ∪ F(w, c)
10 |  |  end
11 |  end
12 end
13 if max_{sense∈S} mean(α[sense]) = 0 then
14 |  if LCB then
15 |  |  return arg max_{(_,cluster)∈S} |cluster|
16 |  else
17 |  |  return −1 // reject to classify
18 |  end
19 else
20 |  return arg max_{(sense,_)∈S} mean(α[sense])
21 end
```

3.5 Word Sense Disambiguation with Induced Word Sense Inventory

To disambiguate a target word t in context, we extract context features C and pass them to Algorithm 1. We use the induced sense inventory I and select the sense that has the largest weighted feature overlap with context features or fall back to the largest cluster back-off when context features C do not match the learned sense representations.

The algorithm starts by retrieving induced sense clusters of the target word (line 1). Next, the method starts to accumulate context feature weights of each $sense$ in $\alpha[sense]$. Each word w in a sense $cluster$ brings all its word-feature counts $F(w, c)$: see lines 5-12. Finally, a $sense$ that maximizes mean weight across all context features is chosen (lines 13-21). Optionally, we can resort to the largest cluster back-off (LCB) strategy in case if no context features match sense representations.

[3] `https://azure.microsoft.com/en-us/services/cognitive-services/search`

Note that the induced inventory I is used as a pivot to aggregate word-feature counts $F(w, c)$ of the words in the $cluster$ in order to build feature representations of each induced $sense$. We assume that the sets of similar words per sense are compatible with each other's context. Thus, we can aggregate ambiguous feature representations of words in a sense cluster. In a way, occurrences of cluster members form the training set for the sense, i.e. contexts of {chair, bed, bench, stool, sofa, desk, cabinet}, add to the representation of "table (furniture)" in the model. Here, ambiguous cluster members like "chair" (which could also mean "chairman") add some noise, but its influence is dwarfed by the aggregation over all cluster members. Besides, it is unlikely that the target ("table") and the cluster member ("chair") share the same homonymy, thus noisy context features hardly play a role when disambiguating the target in context. For instance, for scoring using language model features, we retrieve the context of the target word and substitute the target word one by one of the cluster words. To close the gap between the aggregated dependency per sense $\alpha[sense]$ and dependencies observed in the target's context C, we use the similarity of features: we expand every feature $c \in C$ with 200 of most similar features and use them as additional features (lines 2-4).

We run disambiguation independently for each of the feature types listed above, e.g. dependencies or co-occurrences. Next, independent predictions are combined using the majority-voting rule.

3.6 Interpretability of the Method

Results of disambiguation can be interpreted by humans as illustrated by Figure 2. In particular, our approach is interpretable at three levels:

1. Word sense inventory. To make induced word sense inventories interpretable we display senses of each word as an ego-network of its semantically related words. For instance, the network of the word "table" in our example is constructed from two tightly related groups of words that correspond to "furniture" and "data" senses. These labels of the clusters are obtained automatically (see Section 3.4).

While alternative methods, such as *AdaGram*, can generate sense clusters, our approach makes the senses better interpretable due to hypernyms and image labels that summarize senses.

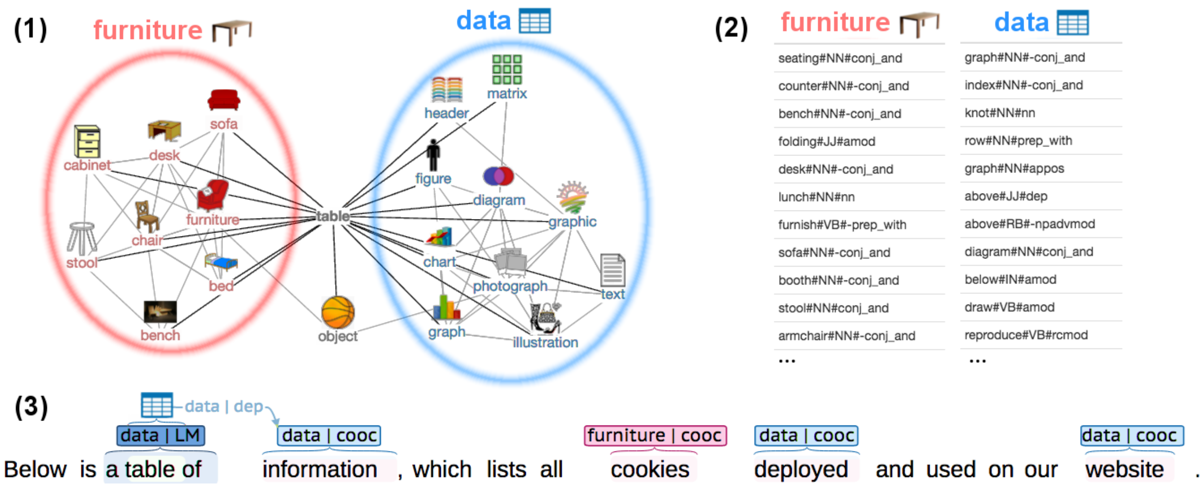

Figure 2: Interpretation of the senses of the word "table" at three levels by our method: (1) word sense inventory; (2) sense feature representation; (3) results of disambiguation in context. The sense labels ("furniture" and "data") are obtained automatically based on cluster labeling with hypernyms. The images associated with the senses are retrieved using a search engine:"table data" and "table furniture".

2. Sense feature representation. Each sense in our model is characterized by a list of sparse features ordered by relevance to the sense. Figure 2 (2) shows most salient dependency features to senses of the word "table". These feature representations are obtained by aggregating features of sense cluster words.

In systems based on dense vector representations, there is no straightforward way to get the most salient features of a sense, which makes the analysis of learned representations problematic.

3. Disambiguation method. To provide the reasons for sense assignment in context, our method highlights the most discriminative context features that caused the prediction. The discriminative power of a feature is defined as the ratio between its weights for different senses.

In Figure 2 (3) words "information", "cookies", "deployed" and "website" are highlighted as they are most discriminative and intuitively indicate on the "data" sense of the word "table" as opposed to the "furniture" sense. The same is observed for other types of features. For instance, the syntactic dependency to the word "information" is specific to the "data" sense.

Alternative unsupervised WSD methods that rely on word sense embeddings make it difficult to explain sense assignment in context due to the use of dense features whose dimensions are not interpretable.

4 Experiments

We use two lexical sample collections suitable for evaluation of unsupervised WSD systems. The first one is the Turk Bootstrap Word Sense Inventory (TWSI) dataset introduced by Biemann (2012). It is used for testing different configurations of our approach. The second collection, the SemEval 2013 word sense induction dataset by Jurgens and Klapaftis (2013), is used to compare our approach to existing systems. In both datasets, to measure WSD performance, induced senses are mapped to gold standard senses. In experiments with the TWSI dataset, the models were trained on the Wikipedia corpus[4] while in experiments with the SemEval datasets models are trained on the ukWaC corpus (Baroni et al., 2009) for a fair comparison with other participants.

4.1 TWSI Dataset

4.1.1 Dataset and Evaluation Metrics

This test collection is based on a crowdsourced resource that comprises 1,012 frequent nouns with 2,333 senses and average polysemy of 2.31 senses per word. For these nouns, 145,140 annotated sentences are provided. Besides, a sense inventory is explicitly provided, where each sense is represented with a list of words that can substitute target noun in a given sentence. The sense distribution across sentences in the dataset is highly

[4]We use a Wikipedia dump from September 2015: `http://doi.org/10.5281/zenodo.229904`

skewed as 79% of contexts are assigned to the most frequent senses. Thus, in addition to the full TWSI dataset, we also use a balanced subset featuring five contexts per sense and 6,166 sentences to assess the quality of the disambiguation mechanism for smaller senses. This dataset contains no monosemous words to completely remove the bias of the most frequent sense. Note that de-biasing the evaluation set does not de-bias the word sense inventory, thus the task becomes harder for the balanced subset.

For the TWSI evaluation, we create an explicit mapping between the system-provided sense inventory and the TWSI word senses: senses are represented as the bag of words, which are compared using cosine similarity. Every induced sense gets assigned at most one TWSI sense. Once the mapping is completed, we calculate Precision, Recall, and F-measure. We use the following baselines to facilitate interpretation of the results: (1) MFS of the TWSI inventory always assigns the most frequent sense in the TWSI dataset; (2) LCB of the induced inventory always assigns the largest sense cluster; (3) Upper bound of the induced vocabulary always selects the correct sense for the context, but only if the mapping exists for this sense; (4) Random sense of the TWSI and the induced inventories.

4.1.2 Discussion of Results

The results of the TWSI evaluation are presented in Table 1. In accordance with prior art in word sense disambiguation, the most frequent sense (MFS) proved to be a strong baseline, reaching an F-score of 0.787, while the random sense over the TWSI inventory drops to 0.536. The upper bound on our induced inventory (F-score of 0.900) shows that the sense mapping technique used prior to evaluation does not drastically distort the evaluation scores. The LCB baseline of the induced inventory achieves an F-score of 0.691, demonstrating the efficiency of the LCB technique.

Let us first consider models based on single features. Dependency features yield the highest precision of 0.728, but have a moderate recall of 0.343 since they rarely match due to their sparsity. The LCB strategy for these rejected contexts helps to improve recall at cost of precision. Co-occurrence features yield significantly lower precision than the dependency-based features, but their recall is higher. Finally, the language model features yield very balanced results in terms of both precision and recall. Yet, the precision of the model based on this feature type is significantly lower than that of dependencies.

Not all combinations improve results, e.g. combination of three types of features yields inferior results as compared to the language model alone. However, a combination of the language model with dependency features does provide an improvement over the single models as both these models bring strong signal of complementary nature about the semantics of the context. The dependency features represent syntactic information, while the LM features represent lexical information. This improvement is even more pronounced in the case of the balanced TWSI dataset. This combined model yields the best F-scores overall.

Table 2 presents the effect of the feature expansion based on the graph of similar features. For a low-recall model such the one based on syntactic dependencies, feature expansion makes a lot of sense: it almost doubles recall, while losing some precision. The gain in F-score using this technique is almost 20 points on the full TWSI dataset. However, the need for such expansion vanishes when two principally different types of features (precise syntactic dependencies and high-coverage trigram language model) are combined. Both precision and F-score of this combined model outperforms that of the dependency-based model with feature expansion by a large margin.

Figure 3 illustrates how granularity of the induced sense inventory influences WSD performance. For this experiment, we constructed three inventories, setting the number of most similar words in the ego-network n to 200, 100 and 50. These settings produced inventories with respectively 1.96, 2.98 and 5.21 average senses per target word. We observe that a higher sense granularity leads to lower F-scores. This can be explained because of (1) the fact that granularity of the TWSI is similar to granularity of the most coarse-grained inventory; (2) the higher the number of senses, the higher the chance to make a wrong sense assignment; (3) due to the reduced size of individual clusters, we get less signal per sense cluster and noise becomes more pronounced.

To summarize, the best precision is reached by a model based on un-expanded dependencies and the best F-score can be obtained by a combination of models based on un-expanded dependency features and language model features.

Model	#Senses	Full TWSI Prec.	Full TWSI Recall	Full TWSI F-score	Sense-Balanced TWSI Prec.	Sense-Balanced TWSI Recall	Sense-Balanced TWSI F-score
MFS of the TWSI inventory	2.31	0.787	0.787	0.787	0.373	0.373	0.373
Random Sense of the TWSI inventory	2.31	0.536	0.534	0.535	0.160	0.160	0.160
Upper bound of the induced inventory	1.96	1.000	0.819	0.900	1.000	0.598	0.748
Largest Cluster Back-Off (LCB) of the induced inventory	1.96	0.691	0.690	0.691	0.371	0.371	0.371
Random sense of the induced inventory	1.96	0.559	0.558	0.558	0.325	0.324	0.324
Dependencies	1.96	**0.728**	0.343	0.466	**0.432**	0.190	0.263
Dependencies + LCB	1.96	0.689	0.680	0.684	0.388	0.385	0.387
Co-occurrences (Cooc)	1.96	0.570	0.563	0.566	0.336	0.333	0.335
Language Model (LM)	1.96	0.685	0.677	0.681	0.416	0.412	0.414
Dependencies + LM + Cooc	1.96	0.644	0.636	0.640	0.388	0.386	0.387
Dependencies + LM	1.96	0.689	**0.681**	**0.685**	0.426	**0.422**	**0.424**

Table 1: WSD performance of different configurations of our method on the full and the sense-balanced TWSI datasets based on the coarse inventory with 1.96 senses/word ($N = 200, n = 200$).

Model	Precision	Recall	F-score	Precision	Recall	F-score
Dependencies	**0.728**	0.343	0.466	**0.432**	0.190	0.263
Dependencies Exp.	0.687	**0.633**	**0.659**	0.414	**0.379**	**0.396**
Dependencies + LM	**0.689**	**0.681**	**0.685**	**0.426**	**0.422**	**0.424**
Dependencies Exp. + LM	0.684	0.676	0.680	0.412	0.408	0.410

Table 2: Effect of the feature expansion: performance on the full (on the left) and the sense-balanced (on the right) TWSI datasets. The models were trained on the Wikipedia corpus using the coarse inventory (1.96 senses per word). The best results overall are underlined.

4.2 SemEval 2013 Task 13 Dataset

4.2.1 Dataset and Evaluation Metrics

The task of word sense induction for graded and non-graded senses provides 20 nouns, 20 verbs and 10 adjectives in WordNet-sense-tagged contexts. It contains 20-100 contexts per word, and 4,664 contexts in total with 6,73 sense per word in average. Participants were asked to cluster instances into groups corresponding to distinct word senses. Instances with multiple senses were labeled with a score between 0 and 1.

Performance is measured with three measures that require a mapping of inventories (Jaccard Index, Tau, WNDCG) and two cluster comparison measures (Fuzzy NMI, Fuzzy B-Cubed).

4.2.2 Discussion of Results

Table 3 presents results of evaluation of the best configuration of our approach trained on the ukWaC corpus. We compare our approach to four SemEval participants and two state-of-the-art systems based on word sense embeddings: *AdaGram* (Bartunov et al., 2016) based on Bayesian stick-breaking process[5] and *SenseGram* (Pelevina et al., 2016) based on clustering of ego-network generated using word embeddings[6]. The *AI-KU* system (Baskaya et al., 2013) directly clusters test contexts using the k-means algorithm based on lexical substitution features. The *Unimelb* system (Lau et al., 2013) uses one hierarchical topic model to induce and disambiguate senses of one word. The *UoS* system (Hope and Keller, 2013) induces senses by building an ego-network of a word using dependency relations, which is subsequently clustered using the MaxMax clustering algorithm. The *La Sapienza* system (Jurgens and Klapaftis, 2013), relies on WordNet for the sense inventory and disambiguation.

In contrast to the TWSI evaluation, the most fine-grained model yields the best scores, yet the inventory of the task is also more fine-grained than the one of the TWSI (7.08 vs. 2.31 avg. senses per word). Our method outperforms the knowledge-based system of *La Sapienza* according to two of three metrics metrics and the *SenseGram* system based on sense embeddings according to four of five metrics. Note that *SenseGram* outperforms all other systems according to the Fuzzy B-Cubed metric, which is maximized in the "All instances, One sense" settings. Thus this result may be due to

[5] https://github.com/sbos/AdaGram.jl

[6] https://github.com/tudarmstadt-lt/sensegram

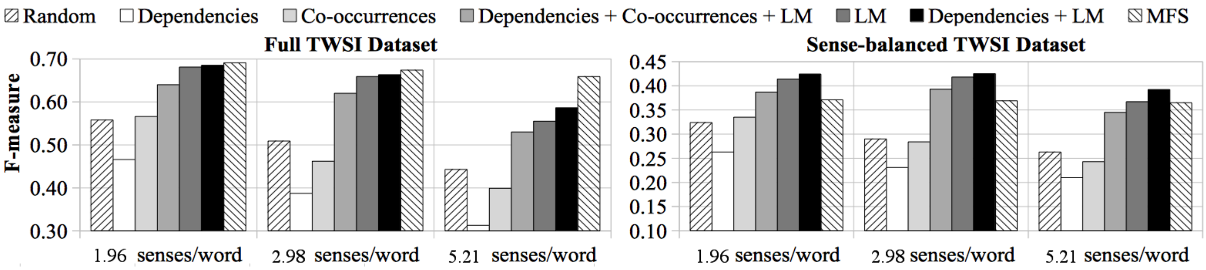

Figure 3: Impact of word sense inventory granularity on WSD performance: the TWSI dataset.

Model	Jacc. Ind.	Tau	WNDCG	Fuzzy NMI	Fuzzy B-Cubed
All Instances, One sense	0.192	0.609	0.288	0.000	0.623
1 sense per instance	0.000	0.953	0.000	0.072	0.000
Most Frequent Sense	0.552	0.560	0.412	–	–
AI-KU	0.197	0.620	**0.387**	**0.065**	0.390
AI-KU (remove5-add1000)	0.245	0.642	0.332	0.039	0.451
Unimelb (50k)	0.213	0.620	0.371	0.060	0.483
UoS (top-3)	0.232	0.625	0.374	0.045	0.448
La Sapienza (2)	0.149	0.510	0.383	–	–
AdaGram, $\alpha = 0.05$, 100 dim. vectors	**0.274**	**0.644**	0.318	0.058	0.470
SenseGram, 100 dim., CBOW, weight, sim., $p = 2$	0.197	0.615	0.291	0.011	**0.615**
Dependencies + LM (1.96 senses/word)	0.239	0.634	0.300	0.041	**0.513**
Dependencies + LM (2.98 senses/word)	0.242	0.634	0.300	0.041	0.504
Dependencies + LM (5.21 senses/word)	**0.253**	**0.638**	**0.300**	**0.041**	0.479

Table 3: WSD performance of the best configuration of our method identified on the TWSI dataset as compared to participants of the SemEval 2013 Task 13 and two systems based on word sense embeddings (AdaGram and SenseGram). All models were trained on the ukWaC corpus.

difference in granularities: the average polysemy of the *SenseGram* model is 1.56, while the polysemy of our models range from 1.96 to 5.21.

Besides, our system performs comparably to the top unsupervised systems participated in the competition: It is on par with the top SemEval submissions (*AI-KU* and *UoS*) and the another system based on embeddings (*AdaGram*), in terms of four out of five metrics (Jaccard Index, Tau, Fuzzy B-Cubed, Fuzzy NMI).

Therefore, we conclude that our system yields comparable results to the state-of-the-art unsupervised systems. Note, however, that none of the rivaling systems has a comparable level of interpretability to our approach. This is where our method is unique in the class of unsupervised methods: feature representations and disambiguation procedure of the neural-based *AdaGram* and *SenseGram* systems cannot be straightforwardly interpreted. Besides, inventories of the existing systems are represented as ranked lists of words lacking features that improve readability, such as hypernyms and images.

5 Conclusion

In this paper, we have presented a novel method for word sense induction and disambiguation that relies on a meta-combination of dependency features with a language model. The majority of existing unsupervised approaches focus on optimizing the accuracy of the method, sacrificing its interpretability due to the use of opaque models, such as neural networks. In contrast, our approach places a focus on interpretability with the help of sparse readable features. While being interpretable at three levels (sense inventory, sense representations and disambiguation), our method is competitive to the state-of-the-art, including two recent approaches based on sense embeddings, in a word sense induction task. Therefore, it is possible to match the performance of accurate, but opaque methods when interpretability matters.

Acknowledgments

We acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) foundation under the JOIN-T project.

References

Eneko Agirre and Philip G. Edmonds. 2007. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science & Business Media.

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145, Mexico City, Mexico. Springer.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS'2016)*, Cadiz, Spain.

Osman Baskaya, Enis Sert, Volkan Cirik, and Deniz Yuret. 2013. AI-KU: Using Substitute Vectors and Co-Occurrence Modeling for Word Sense Induction and Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 300–306, Atlanta, GA, USA. Association for Computational Linguistics.

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

Chris Biemann. 2006. Chinese Whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the first workshop on graph based methods for natural language processing*, pages 73–80, New York City, NY, USA. Association for Computational Linguistics.

Chris Biemann. 2012. Turk Bootstrap Word Sense Inventory 2.0: A Large-Scale Resource for Lexical Substitution. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 4038–4042, Istanbul, Turkey. European Language Resources Association.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. NASARI: a novel approach to a semantically-aware representation of items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 567–577, Denver, CO, USA. Association for Computational Linguistics.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D. Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th Language Resources and Evaluation Conference (LREC'2006)*, pages 449–454, Genova, Italy. European Language Resources Association.

Antonio Di Marco and Roberto Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754.

Beate Dorow, Dominic Widdows, Katarina Ling, Jean-Pierre Eckmann, Danilo Sergi, and Elisha Moses. 2005. Using curvature and markov clustering in graphs for lexical acquisition and word sense discrimination. In *Proceedings of the Meaning-2005 Workshop*, Trento, Italy.

Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19:61–74.

Martin Everett and Stephen P. Borgatti. 2005. Ego network betweenness. *Social networks*, 27(1):31–38.

Stefan Evert. 2005. *The Statistics of Word Cooccurrences Word Pairs and Collocations*. Ph.D. thesis, University of Stuttgart.

Stefano Faralli and Roberto Navigli. 2012. A new minimally-supervised framework for domain word sense disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1411–1422, Jeju Island, Korea, July. Association for Computational Linguistics.

Alex A Freitas. 2014. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

David Hope and Bill Keller. 2013. MaxMax: A Graph-based Soft Clustering Algorithm Applied to Word Sense Induction. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 368–381, Samos, Greece. Springer.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'2012)*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL'2015)*, pages 95–105, Beijing, China. Association for Computational Linguistics.

Nancy Ide and Jean Véronis. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, 24(1):2–40.

David Jurgens and Ioannis Klapaftis. 2013. Semeval-2013 Task 13: Word Sense Induction for Graded and Non-graded Senses. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval'2013)*, pages 290–299, Montreal, Canada. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

Dan Klein, Kristina Toutanova, H. Tolga Ilhan, Sepandar D. Kamvar, and Christopher D. Manning. 2002. Combining Heterogeneous Classifiers for Word-Sense Disambiguation. In *Proceedings of the ACL'2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, volume 8, pages 74–80, Philadelphia, PA, USA. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95)*, volume 1, pages 181–184, Detroit, MI, USA. IEEE.

Jey Han Lau, Paul Cook, and Timothy Baldwin. 2013. unimelb: Topic Modelling-based Word Sense Induction. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 307–311, Atlanta, GA, USA. Association for Computational Linguistics.

Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'2002)*, volume 10, pages 41–48, Philadelphia, PA, USA. Association for Computational Linguistics.

Michael Lesk. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, Toronto, ON, Canada. ACM.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, MD, USA. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Conference on Empirical Methods in Natural Language Processing (EMNLP'2015)*, pages 1722–1732, Lisboa, Portugal. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, CA, USA. Association for Computational Linguistics.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML'1998)*, volume 98, pages 296–304, Madison, WI, USA. Morgan Kaufmann Publishers Inc.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Workshop at International Conference on Learning Representations (ICLR)*, pages 1310–1318, Scottsdale, AZ, USA.

Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych. 2012. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 1781–1796, Mumbai, India. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 Task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, CO, USA. Association for Computational Linguistics.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics*, pages 216–225, Uppsala, Sweden.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Hwee Tou Ng. 1997. Exemplar-Based Word Sense Disambiguation: Some Recent Improvements. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 208–213, Providence, RI, USA. Association for Computational Linguistics.

Luis Nieto Piña and Richard Johansson. 2016. Embedding senses for efficient graph-based word sense disambiguation. In *Proceedings of TextGraphs-10: the Workshop on Graph-based Methods for Natural Language Processing*, pages 1–5, San Diego, CA, USA. Association for Computational Linguistics.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Alexander Panchenko and Olga Morozova. 2012. A study of hybrid similarity measures for semantic relation extraction. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 10–18, Avignon, France. Association for Computational Linguistics.

Alexander Panchenko. 2016. Best of both worlds: Making word sense embeddings interpretable. In *Proceedings of the 10th Language Resources and Evaluation Conference (LREC'2016)*, pages 2649–2655, Portoro, Slovenia. European Language Resources Association (ELRA).

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 613–619, Edmonton, AB, Canada.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2016. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv preprint arXiv:1612.04757*.

Ted Pedersen and Rebecca Bruce. 1997. Distinguishing word senses in untagged text. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP'1997)*, pages 197–207, Providence, RI, USA. Association for Computational Linguistics.

Ted Pedersen, Satanjeev Banerjee, and Siddharth Patwardhan. 2005. Maximizing semantic relatedness to perform word sense disambiguation. *University of Minnesota supercomputing institute research report UMSI*, 25:2005.

Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. 2016. Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, Berlin, Germany. Association for Computational Linguistics.

Martin Riedl. 2016. *Unsupervised Methods for Learning and Using Semantics of Natural Language*. Ph.D. thesis, Technische Universität Darmstadt, Darmstadt.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China. Association for Computational Linguistics.

Eugen Ruppert, Manuel Kaufmann, Martin Riedl, and Chris Biemann. 2015. Jobimviz: A web-based visualization for graph-based distributional semantic models. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 103–108, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Hinrich Schütze. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'2014)*, pages 151–160, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Tim Van de Cruys. 2010. Mining for meaning: The extraction of lexicosemantic knowledge from text. *Groningen Dissertations in Linguistics*, 82.

Stijn Van Dongen. 2008. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141.

Alfredo Vellido, José David Martín, Fabrice Rossi, and Paulo J.G. Lisboa. 2011. Seeing is believing: The importance of visualization in real-world machine learning applications. In *Proceedings of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'2011)*, pages 219–226, Bruges, Belgium.

Alfredo Vellido, José D. Martín-Guerrero, and Paulo J.G. Lisboa. 2012. Making machine learning models interpretable. In *20th European Symposium on Artificial Neural Networks, ESANN*, volume 12, pages 163–172, Bruges, Belgium.

Jean Véronis. 2004. HyperLex: Lexical cartography for information retrieval. *Computer Speech and Language*, 18:223–252.

Heng Low Wee. 2010. Word Sense Prediction Using Decision Trees. Technical report, Department of Computer Science, National University of Singapore.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'2002)*, pages 1–7, Taipei, Taiwan. Association for Computational Linguistics.

Deniz Yuret. 2012. FASTSUBS: An efficient and exact procedure for finding the most likely lexical substitutes based on an n-gram language model. *IEEE Signal Processing Letters*, 19(11):725–728.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. Association for Computational Linguistics.