Noun Sense Induction and Disambiguation using Graph-Based Distributional Semantics

Alexander Panchenko, Johannes Simon, Martin Riedl and Chris Biemann
Technische Universität Darmstadt, Computer Science Department, LT Group
Hochschulstr. 10, Darmstadt, Germany
{panchenko,simon,riedl,biem}@lt.informatik.tu-darmstadt.de

Abstract

We introduce an approach to unsupervised word sense induction and disambiguation: sense representations for ambiguous words are learned from distributional evidence and subsequently used to disambiguate word instances in context. These sense representations obtained by clustering dependency-parse-based second-order similarity networks as a pivot. We then add features for disambiguation from heterogeneous sources such as window-based and sentence-wide co-occurrences, and explore various schemes to combine these complementary context clues. Our method reaches a performance comparable to the state-of-the-art unsupervised word sense disambiguation systems including top participants of the SemEval 2013 word sense induction task and a more recent state-of-the-art neural word sense induction system.

1 Introduction

A word sense disambiguation (WSD) system takes as input a word and its context and outputs a sense of this word (Navigli, 2009). While the goal of all such methods is the same, there are substantial differences in their implementation. Some systems use knowledge-based approaches that rely on hand-crafted sense inventories, such as WordNet (Miller, 1995), while others use supervised approaches that learn from hand-labeled training data, such as SemCor (Miller et al., 1993). However, hand-crafted lexical resources and training data are expensive to create, often inconsistent and domain-dependent. Furthermore, these methods assume a fixed sense inventory for each word. This is problematic as (1) senses emerge and disappear over time; (2) different applications require different granularities of a sense inventory.

An alternative route explored in this paper is based on unsupervised knowledge-free approach. Our method learns an interpretable sense inventory by clustering semantically similar words. To learn sense inventories, we rely on the JoBimText framework and distributional semantics (Biemann and Riedl, 2013), adding a word sense disambiguation functionality on the top of it.

The key contribution of this paper is a framework that relies on such induced inventories as a pivot for learning contextual feature representations and uses them for disambiguation. The advantage of our method, compared to prior art, is that it can incorporate several types of context features in an unsupervised way. We demonstrate our approach, which combines four heterogeneous types of context features and achieves state of the art results in unsupervised WSD.

2 Related Work

Approaches to WSD vary according to the level of supervision and according to the amount of external knowledge they use (Agirre and Edmonds, 2007; Navigli, 2009).

Supervised approaches use an explicitly sense-labeled training corpus to construct a model, usually building one model per target word. Successful machine learning setups include SVMs (Lee and Ng, 2002) and classifier ensembles (Klein et al., 2002). Wee (2010) shows that decision trees using bag-of-word features are unable to outperform the most frequent sense baseline. Supervised approaches achieve the top performance in shared tasks on WSD such as SemEval, but require considerable amounts of sense-labeled examples.

A WSD method that uses predefined dictionaries, lexical resources or semantic ontologies can be considered *knowledge-based*. Knowledge-based systems rely on a lexical resource and vary from the classical Lesk (1986) algorithm that use word definitions to the *BabelFy* (Moro et al., 2014) system

that harnesses a multilingual semi-automatically constructed lexical semantic network. Knowledge-based approaches to WSD do not learn a model per target, but rather utilize information from a lexical resource that provides the sense inventory as well. Examples include (Lesk, 1986; Banerjee and Pedersen, 2002; Pedersen et al., 2005).

In this paper we deal with *unsupervised* and *knowledge-free* WSD approaches. They use neither handcrafted lexical resources nor hand-annotated sense-labeled corpora. Instead, they induce word sense inventories automatically from corpora. According to Navigli (2009), unsupervised WSD methods fall into two categories: context clustering (Pedersen and Bruce, 1997; Schütze, 1998) and word (ego-network) clustering (Lin, 1998; Pantel and Lin, 2002; Widdows and Dorow, 2002; Biemann, 2006; Hope and Keller, 2013a).

Context clustering methods, e.g. (Schütze, 1998), usually represent an instance 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 popular skip-gram model (Mikolov et al., 2013) also belong to the same group. They learn one embedding word vector per 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).

The *AI-KU* system (Baskaya et al., 2013) is also based on context clustering. First, for each instance the system identifies the 100 most probable lexical substitutes using an *n*-gram model (Yuret, 2012). Each instance is thus represented by a bag of substitutes. These vectors are clustered using *k*-means. The *Unimelb* system by Lau et al. (2013) implements context clustering using the Hierarchical Dirichlet Process (HDP) (Teh et al., 2006). Latent topics discovered in the training instances, specific to every word, are interpreted as word senses.

Another class of word sense induction systems cluster *word ego-networks*, rather than single instances of words. 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, cf. Figure 1. Nodes of an 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, 2013a). Edges usually represent semantic similarities resp. association strength between nodes. The sense induction process using word graphs was previously explored by (Widdows and Dorow, 2002; Biemann, 2010; Hope and Keller, 2013a). 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, similar to the simplified Lesk algorithm.

The *UoS* system by Hope and Keller (2013a) builds a word ego-network with nodes being the 300 highest-ranked words in a dependency relation with the target word and clusters the graph to obtain senses weighted by word similarities. The graph is clustered with the MaxMax algorithm. Similar clusters are merged. 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.

While arguably the *UoS* system is the most similar to ours, there are crucial differences. First, nodes in their ego network are (first-order) context features, not (second-order) similar words. Second, edge weights in our network represent the number of shared features, not the significance of co-occurrences. Finally, their disambiguation component relies on overlap between context and a sense's cluster words.

Our system combines several of the above ideas, such as word sense induction based on clustering word similarities (Pantel and Lin, 2002), but in contrast to other unsupervised knowledge-free systems, we are able to combine and systematically evaluate the evidence from several features that model context differently.

3 Data-Driven Noun Sense Modelling

Our method consists of the three steps: computation of a distributional thesaurus, word sense induction, and building a disambiguation model of the induced senses.

3.1 Distributional Thesaurus of Nouns

The goal of this step is to build a graph of word similarities, such as "(tablet, notebook, 0.781)".[1] To compute the graph, we used the *JoBimText* framework (Biemann and Riedl, 2013). While multiple alternatives exist for the computation of semantic similarity e.g. (Mikolov et al., 2013), this framework is convenient in our case due to efficient

[1] We use the terms "semantic similarity/relatedness" to denote scores derived with a distributional semantics approach.

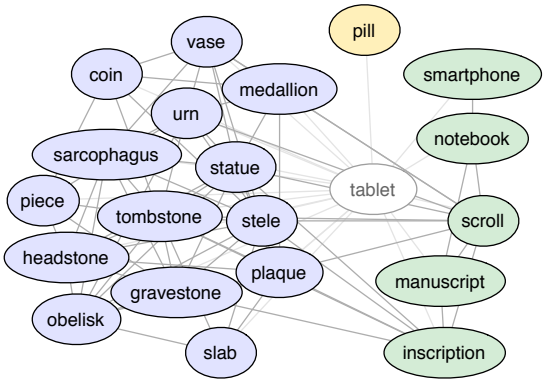

Figure 1: Visualization of the ego-network of the word "tablet" with three color-coded senses: "stone", "device", and "pill". Note that the ego word "tablet" is excluded from clustering.

computation of nearest neighbours for all words in the corpus while providing comparable precision (Riedl, 2016). For each noun in the corpus we retain the 200 most similar nouns.

3.2 Noun Sense Induction

Similar to (Pantel and Lin, 2002) and (Biemann, 2006), we induce a sense inventory which represents senses with word clusters. For instance, the the sense "tablet (device)" can be represented by the cluster "smartphone, notebook, scroll, manuscript, inscription", see Figure 1. To compute the clustering, first we construct an ego-network G of a word t and then perform graph clustering of this network. An ego-network (Everett and Borgatti, 2005) contains all nodes connected to the target node, called "ego". The identified clusters are interpreted as senses. Figure 1 depicts an ego-network of "tablet". Panchenko et al. (2013) proposed a system for dynamic visualization of word ego-networks similar to those used in our method.[2] The key property of word ego-networks is that the words with similar senses tend to be connected among each other, while having fewer connections to words from other senses, therefore forming clusters.

The sense induction processes one word t of the distributional thesaurus T per iteration. First, we retrieve nodes of the ego-network G being the N most similar words V of t according to T. Note that the target word t itself is not part of the ego-network. Second, we connect the nodes in G to their n most similar words from T. Finally, the ego-network is clustered with the Chinese Whispers algorithm (Biemann, 2006).

The sense induction algorithm has two meta-parameters: the *ego-network size* (N) of a target ego word t; and the *ego-network connectivity* (n) one of these neighbors v is allowed to have within the network. The parameter n regulates the granularity of the inventory. In our experiments we set N and n to 200 to obtain a coarse-grained inventory. In preliminary experiments, we found inventories based on dependency features superior to other inventories, which is why we use only dependency-based similarities in our WSI experiments.

3.3 Disambiguation of Induced Noun Senses

The goal of this step is to construct a disambiguation model $P(s_i|C)$ for each of the induced senses $s_i \in S$, where C is a feature representation of the target word w in a context. We approximate the conditional probability of the sense s_i in the context $C = \{c_1, ..., c_m\}$ with the Naïve Bayes model:

$$P(s_i|C) = \frac{P(s_i)\prod_{j=1}^{|C|} P(c_j|s_i)}{P(c_1, ..., c_m)}, \qquad (1)$$

where the best sense given C is chosen as following: $s_i^* = \arg\max_{s_i} P(s_i)\prod_{j=1}^{|C|} P(c_j|s_i)$.

To learn this model we use the assumption that words from a sense cluster S are, to some extent, semantically substitutable. For example, consider the sense cluster that represents the "fish" sense of the word "bass": {trout, catfish, eel, perch} and the following sentence: "*Most fish such as • live in freshwater lakes and rivers*". As can be observed in this example, similar words usually occur in similar contexts and thus often have similar context features. As it will be clear from our experiments, in spite of inherent noise in such training data one can use these data for training a disambiguation model.

Based on this assumption, it is possible to extract sense representations by aggregation of features from all words of the cluster s_i: we simply count in the training corpus the number of co-occurrences $f(w_k, c_j)$ and the cluster word w_{ik} with the context feature c_j across all words belonging to the sense cluster s_i: $\{w_1, ..., w_n\}$.

We cannot directly count any sense frequencies $f(s_i)$ or joint sense-feature frequencies $f(s_i, c_j)$ from an unlabeled text corpus. To estimate these frequencies we utilize an implication of our hypothesis: since two similar words are assumed to

[2] http://www.serelex.org

be substitutable, we assume any occurrence of the i-th word from the k-th cluster, denoted as w_k, to be interchangeable with an occurrence of sense s_i. The frequency of s_i is then given by $f(s_i) = \sum_i^{|s_i|} f(w_k)$, where $|s_i|$ is the number of words in the sense cluster s_i. The same principle can be applied to determine a joint frequency $f(s_i, c_j)$. To estimate the probability of a sense feature given a cluster word, we normalize the joint frequency by word frequency. This solves the problem of dominating high frequency cluster words:

$$P(c_j|w_k) = \frac{f(w_k, c_j)}{f(w_k)}. \quad (2)$$

A sense cluster usually contains a large number of similar words (up to $N = 200$ in our case). Often there is a high discrepancy among the similarities of the cluster words to the target word. Thus, some words better represent the sense than the others. To account for this effect, we introduce an additional weighting coefficient λ_k that is equal to the similarity between k-th cluster word w_k and the target word w being disambiguated.

While cluster words may be ambiguous, this issue is compensated by the fact that most cluster words have common features, while the noisy features of ambiguous words are specific to these words: they are not confirmed by noisy features of other ambiguous words. In some cases this assumption does not hold. For instance, the word "Chelsea" is similar to other words such as "Milan" or "Barcelona" that can represent both either a club or a city.

To normalize the score we divide it by the sum of all the weights $\Lambda_i = \sum_k^{|s_i|} \lambda_k$:

$$P(c_j|s_i) = \frac{1-\alpha}{\Lambda_i} \sum_k^{|s_i|} \lambda_k \frac{f(w_k, c_j)}{f(w_k)} + \alpha, \quad (3)$$

where α is a small number, e.g. 10^{-5}, added for smoothing.

The prior probability of each sense is computed based on the largest cluster heuristic:

$$P(s_i) = \frac{|s_i|}{\sum_{s_i \in S} |s_i|}. \quad (4)$$

We also explored estimation of the prior by a weighted average of cluster word counts, but this method provided lower results:

$$P(s_i) = \frac{1}{\Lambda_i} \sum_k^{|s_i|} \lambda_k \, f(w_k). \quad (5)$$

Note that to calculate the sense models we only need (1) the distributional thesaurus T; (2) sense clusters; and (3) word-feature frequencies: $f(w_k) = f_{n*}$, and $f(w_k, c_j) = f_{nm}$, where n is the index of the word w_k and m is the index of the feature c_j in a word-feature matrix. Finally, sense features are pruned: in our experiments, each sense s_i is represented with most significant 20,000 context features in terms of $P(c_j|s_i)$.

3.4 Feature Extraction and Combination

Our method learns separate models $P(s_i|C)$ for each type of context features. During classification, we either use these single-featured models directly or combine them at the feature- or meta-levels as described below.

Single features. We use four groups of word-feature counts $f(w_k, c_j)$ listed below to estimate probability of the feature given a sense $\hat{P}(c_j|s_i)$. A single-sense model is then trained for each of these feature types. Note that our framework allow using of any other context features if one can estimate $f(w_k, c_j)$ for it.

- **Cluster features** directly use words from the induced sense clusters i.e., the $\hat{P}(c_j|s_i)$ equals to the similarity score λ_{kj} between the target word w_k and the context word c_j.
- **Dependency features** of a target word w_k are all syntactic dependencies attached to it. For instance, the word "tablet" has features such as "subj(•,type)" or "amod(digital,•)", where "•" represents position of the target word. During disambiguation, we use this kind of features in two modes: the first one, denoted as *Deptarget*, represents the context C as a set of all dependencies attached to the target word being disambiguated; the second mode, denoted as *Depall* represents the context C with dependencies of all words in the sentence, not just the target word. This is an expansion of feature representation aiming to compensate the sparsity of the dependency representation.
- **Dependency word features**, denoted as *Depword*, are extracted from all syntactic dependencies attached to a target word w_k. Namely, we reduce dependency features to dependent words. For instance, the feature "subj(•,write)" would result in the feature "write". We also experimented with word co-occurrences, but they provided lower results.
- **Trigram features** are pairs of left and right

words around the target word w_k. For instance, the word "tablet" has features such as "typing_•_or" and "digital_•_.". Similarly to the dependency features, during disambiguation we use two modes to build the context C: the *Trigramtarget* represents the target word with one trigram extracted from its context; the *Trigramall* represents the target word with trigrams extracted from all words in the sentence.

Feature-level combination of features. This method builds the set of context features C uniting different context features under combination, such as dependencies and trigrams. Next, we use the Naïve Bayes model based on this extended context representation to estimate $\hat{P}(s_i|C)$, using conditional probabilities $\hat{P}(c_j|s_i)$ depending on the type of the corresponding feature $c_j \in C$.

Meta-level combination of features. This method starts by performing independent sense classifications with the combined models. Next, these predictions are aggregated using one of the three following strategies:

- **Majority** selects the sense s_i selected by the largest number of single models.
- **Ranks**. First, results of single model classification are ranked by their confidence $\hat{P}(s_i|C)$: the most suitable sense to the context obtains rank one and so on. Next, we assigns the sense with the least sum of ranks.
- **Sum**. This strategy assigns the sense with the largest sum of classification confidences i.e., $\sum_i \hat{P}(s_i|C_k^i)$, where i is the number of the single model.

4 Results

We evaluate our method on three complementary datasets: (1) a small-scale collection of homonyms used for convenient interpretation of results; (2) a large-scale collection of homonyms and polysemous senses used for development of meta-parameters; and (3) a mid-scale SemEval dataset used for comparison with other systems.

In the experiments described below, we trained models on two corpora commonly used for training distributional models: ukWaC (Ferraresi et al., 2008) and Wikipedia[3]. Table 1 presents statistics about these two text collections.

[3]We used a dump of Wikipedia of October 2015: http://panchenko.me/data/joint/corpora/en59g/wikipedia.txt.gz

	# Tokens	Size	Text Type
Wikipedia	$1.863 \cdot 10^9$	11.79 Gb	encyclopaedic
ukWaC	$1.980 \cdot 10^9$	12.05 Gb	Web pages

Table 1: Corpora used for training our models.

4.1 Evaluation on PRJ

The goal of this evaluation is to make sure the method performs as expected in simple settings i.e., in case of homonyms. We chosen a small scale dataset to be able to track each misclassified context.

Dataset. This dataset consists of 60 contexts of words "python", "ruby" and "jaguar", hence the name of the dataset (PRJ). Each word has two homonymous senses, respectively "snake" or "programming language", "gem" or "programming language", and "animal" or "car", respectively. Contexts were randomly sampled from the first three paragraphs of the corresponding Wikipedia articles. Each sense is represented with 10 contexts. We manually assigned senses from the induced inventory derived from the ukWaC corpus. In this experiment, we used the model trained on the ukWaC corpus.

Evaluation metrics. Since the contexts are labeled with the induced senses, we directly use precision and recall without mapping of inventories.

Discussion of results. Agirre and Soroa (2007) suggest that the WSD of homonyms is almost solved problem for supervised systems, reaching F-scores above 0.90. Our results summarized in Table 2 confirm this for the unsupervised approach. Our method reaches precision up to 0.953 and F-score of 0.950.

The three misclassified samples by the system reached F-score of 0.950 are the following. The first one is from the article about "ruby (gem)" which describes possible colors of ruby gems. It was wrongly labeled with the "ruby (color)" sense. The second misclassified example from the "jaguar (animal)" article contains multiple named entities, such as "USA" that strongly related to economic activities such car production. Finally, the reason of misclassification of the third context from the "python (snake)" article is that the "molurus" feature received high score in the "language" sense. We attribute this learning error due to unbalanced

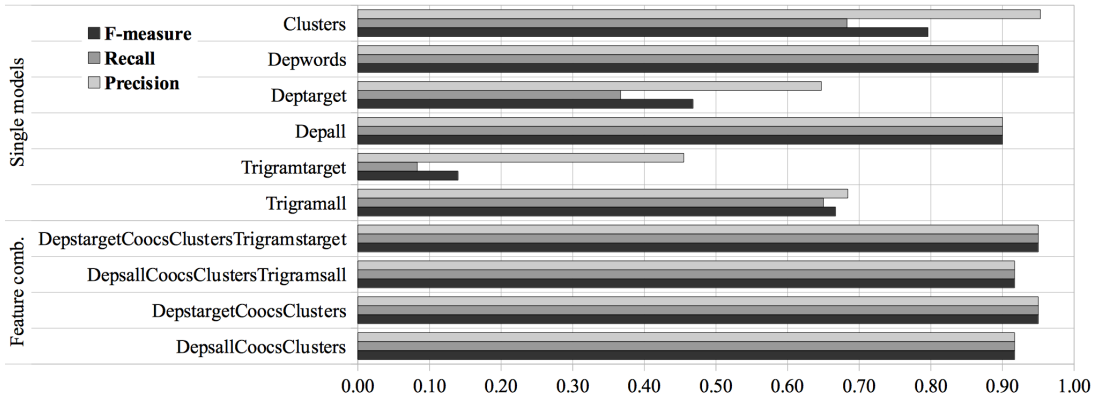

Figure 2: Performance of our method on the PRJ dataset. The models based on meta-combinations are not shown for brevity as they did not improve performance of the presented here models in terms of F-score.

nature of the ukWaC, as in the model trained on Wikipedia this feature has a higher score for the "snake" sense. Thus, we conclude that our approach performs as expected in simple cases, yielding almost no errors.

Combinations of the single predictors neither provide extra improvement in these simple settings: none of the combined models improve the overall results, nor they introduce any extra errors (see Figure 2). Finally, the meta-combination based on sum of ranks yielded the highest precision at the cost of a recall drop (not shown in Figure 2 for brevity).

4.2 Evaluation on TWSI

The goal of this evaluation is to test performance of our method on a large scale dataset that contains both homonyms and polysemous senses.

Dataset. This test collection is based on a large-scale crowdsourced resource (Biemann, 2012) that comprises 1,012 frequent nouns with average polysemy of 2.33 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 skewed resulting in 79% of contexts assigned to the most frequent senses.

Evaluation metrics. To compute performance we create an explicit mapping between the system-provided sense inventory and the TWSI senses: senses are represented as bag of words vectors, which are compared using cosine similarity. Every induced sense gets assigned at most one TWSI sense. Once the mapping is completed, we can calculate Precision and recall of the sense labeling with respect to the original TWSI labeling.

Note that performance of a disambiguation model depends on quality of the sense mapping. Therefore, we use five baselines that facilitate interpretation of the results:

1. **MFS of the TWSI inventory** assigns the most frequent sense in the TWSI dataset.
2. **Random sense of the TWSI inventory**.
3. **MFS of the induced inventory** assigns the identifier of the largest sense cluster.
4. **Upper bound of the induced vocabulary** selects the correct sense for the context, but only if the mapping exist for this sense.
5. **Random sense of the induced inventory**.

Discussion of results. Table 2 presents evaluation of our method trained on the Wikipedia corpus (comparison of these results with the ukWaC corpus is provided in Figure 3). First, one can observe that, similarly to the PRJ dataset, the *Cluster* features yield a precise results up to $P = 0.719$. Yet, recall of these feature is inherently limited by the size of these clusters (15 to 200 words as compared to up to 20,000 for other types of features). Besides, *Trigramtarget* features yield even higher precision of 0.729, but their recall of 0.193 is even less than that of clusters. The single model based on the *Deptarget* features balances precision and recall, reaching F-measure of 0.571 at $P = 0.709$.

Several models based on feature- and meta-level combinations clearly outperform single-feature models. The best scores in terms of F-score (0.696-0.698) are obtained by a combination of four fea-

ture types (*Deptarget, Depword, Cluster, Trigramtarget*) at the feature level or using the sum meta-combination. Similar results (F-score of 0.694-0.695) can be obtained via combination of the same features without the *Trigramtarget*. In terms of precision, the best results are delivered by a meta-combination of the above-mentioned features, combined by summing their ranks. In these settings, the combined models yield precision of 0.713-0.720.

Figure 3 compares the performance of our models trained on the Wikipedia corpus and the ukWaC corpus. The Wikipedia-based models consistently outperform their counterparts trained on the ukWaC. This can be attributed to the fact that the TWSI contexts were originally sampled from the Wikipedia. Besides, Wikipedia is a more balanced and "clean" corpus than ukWaC.

All our models outperform the random sense baselines and the most frequent sense (MFS) baseline of the induced inventory in terms of precision and most of them outperforms these baselines in terms of F-score. These results show that the features used in our technique indeed provide a strong signal for word sense disambiguation. However, none of our models was able to outperform the most frequent sense of the TWSI.

We assumed that this is due to the highly skewed nature of the dataset where 79% of contexts are associated with the most frequent sense. To validate the hypothesis that our system yields state-of-the-art performance in spite of this result we compared its performance to a recent unsupervised WSD system based on sense embeddings, called AdaGram (Bartunov et al., 2016). This is a multi-prototype extension of the Skip-gram model (Mikolov et al., 2013), which relies on Bayesian inference to perform sense disambiguation. We chosen this method as it yields state-of-the-art results, outperforming other approaches based on sense embeddings, such as (Neelakantan et al., 2014). We tried several models varying the α parameter that controls granularity of the induced sense inventory. The best AdaGram configuration with the $\alpha =$ equals 0.05 yields F-score on of 0.656, which is below the most frequent sense of the TWSI, similarly to our top model *Deptarget-DepwordClusterTrigramtarget* that reaches F-score of 0.698.

4.3 Evaluation on SemEval-2013 Task 13

The goal of this evaluation is to compare performance of our method to the state-of-the-art unsupervised WSD systems.

Dataset. The SemEval-2013 task 13 "Word Sense Induction for Graded and Non-Graded Senses" (Jurgens and Klapaftis, 2013) 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, which were drawn from the Open American National Corpus. In our experiments, we use the 1,848 noun-based contexts. Participants were asked to cluster these 4,664 instances into groups, with each group corresponding to a distinct word sense. We report result on the 20 nouns as our approach is designed for nouns.

Evaluation metrics. Performance is measured with three measures that require a mapping of sense inventories (Jaccard Index, Tau and WNDCG) and two cluster comparison measures (Fuzzy NMI and Fuzzy B-Cubed).[4] During evaluation the test data is divided into five segments: four of which are used to build the mapping, and one for evaluation.

Discussion of results. Participating teams in this task were *AI-KU* (Baskaya et al., 2013), *Unimelb* (Lau et al., 2013), *UoS* (Hope and Keller, 2013b) and *La Sapienza*. The latter relies on WordNet as sense inventory and uses a knowledge-rich approach to disambiguation. Only the *UoS* used an induced sense inventory, similarly to us, while all other participating teams performed sense clustering directly on the disambiguation instances, thus not being able to classify additional instances without re-clustering the whole dataset.

Table 3 compares the performance of our method to other approaches. As one may observe, most of the combined models only sightly improve over the single-feature models according to Jaccard Index and Fuzzy NMI. However, one class of combined models that achieves a consistent improvement over the single-feature systems is the meta-combination based on the sum of ranks. Similarly to the TWSI experiment, the two best combined models are based either on four (*Deptarget, Depword, Cluster, Trigramtarget*) or three (*Deptarget, Depword, Cluster*) features. These two models

[4] Detailed interpretation of the five performance metrics: https://www.cs.york.ac.uk/semeval-2013/task13/index.php%3Fid=results.html

Model		#Senses	Precision	Recall	F-score
TWSI baselines	MFS of the TWSI inventory	2.31	0.787	0.787	0.787
	Random sense of the TWSI inventory	2.31	0.535	0.535	0.535
Induced baselines	Upper bound of the induced inventory	1.64	1.000	0.746	0.855
	MFS of the induced inventory	1.64	0.642	0.642	0.642
	Random Sense of the induced inventory	1.64	0.559	0.558	0.558
Sense embeddings	AdaGram, $\alpha = 0.05$, upper bound of induced inv.	4.33	1.000	0.865	0.928
	AdaGram, $\alpha = 0.05$	4.33	0.656	0.656	0.656
Single models	Cluster	1.64	**0.719**	0.405	0.518
	Depword	1.64	0.684	0.684	0.684
	Deptarget	1.64	0.709	0.571	0.633
	Depall	1.64	0.689	0.689	0.689
	Trigramtarget	1.64	**0.729**	0.193	0.305
	Trigramall	1.64	0.670	0.561	0.611
Feature comb.	DeptargetDepwordClusterTrigramtarget	1.64	0.698	**0.698**	**0.698**
	DepallDepwordClusterTrigramall	1.64	0.697	**0.697**	**0.697**
	DeptargetDepword Cluster	1.64	0.694	**0.694**	**0.694**
	DepallDepwordCluster	1.64	0.691	**0.691**	0.691
Meta comb.	Cluster+Deptarget+Depword+Trigramtarget: majority	1.64	**0.718**	0.605	0.656
	Cluster+Deptarget+Depword+Trigramtarget: ranks	1.64	0.687	0.360	0.472
	Cluster+Deptarget+Depword+Trigramtarget: sum	1.64	0.696	**0.696**	**0.696**
	Cluster+Depall+Depword+Trigramall: majority	1.64	0.692	0.685	0.688
	Cluster+Depall+Depword+Trigramall: ranks	1.64	**0.715**	0.420	0.529
	Cluster+Depall+Depword+Trigramall: sum	1.64	0.693	0.693	0.693
	Cluster+Deptarget+Depword: majority	1.64	0.704	0.630	0.665
	Cluster+Deptarget+Depword: ranks	1.64	0.713	0.410	0.521
	Cluster+Deptarget+Depword: sum	1.64	0.695	0.695	**0.695**
	Cluster+Depall+Depword: majority	1.64	0.689	0.688	0.688
	Cluster+Depall+Depword: ranks	1.64	**0.720**	0.406	0.519
	Cluster+Depall+Depword: sum	1.64	0.693	0.693	0.693

Table 2: Performance of our method on the TWSI dataset trained on the Wikipedia corpus. Top 5 scores of our approach per section are set in boldface; the best scores are underlined.

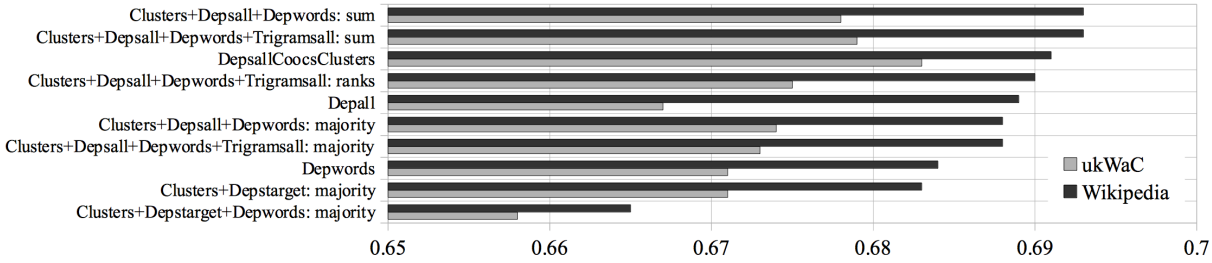

Figure 3: Effect of the corpus choice on the WSD performance: 10 best models according to the F-score on the TWSI dataset trained on Wikipedia and ukWaC corpora.

perform comparably to the best participants of the SemEval challenge or outperform them, depending on the metric. On one hand, the top SemEval system (AI-KU remove5-add1000) reaches Jaccard Index of 0.229 while our approach obtains scores up to 0.219. The second best SemEval system according to this metric (UoS top-3) has a score of 0.220. On the other hand, according to the Tau and Fuzzy B-Cubed scores, our best systems outperform the SemEval participants. Therefore, we conclude that performance of our approach is comparable to the other unsupervised state-of-the-art word sense disambiguation approaches.

Finally, note that none of the unsupervised WSD methods discussed in this paper, including the top-ranked SemEval submissions and the AdaGram, were able to beat the most frequent sense baselines of the respective datasets. Similar results are observed for other recently proposed unsupervised word sense disambiguation methods (Nieto Piña and Johansson, 2016).

5 Conclusions

Performance of the state-of-the-art knowledge-based and supervised WSD systems reached satisfactory levels, but they inherently suffer from inevitable out of vocabulary terms in any "non-standard" domain or language. We presented a new unsupervised knowledge-free approach to word sense induction and disambiguation that addresses these problems as it can be trained on a domain-

Model		Jacc. Ind.	Tau	WNDCG	Fuzzy NMI	Fuzzy B-Cubed
Baselines	One sense for all	0.171	0.627	0.302	0.000	0.631
	One sense per instance	0.000	0.953	0.000	0.072	0.000
	Most Frequent Sense (MFS)	0.579	0.583	0.431	–	–
SemEval systems	AI-KU (add1000)	0.176	0.609	0.205	0.033	0.317
	AI-KU	0.176	0.619	0.393	**0.066**	0.382
	AI-KU (remove5-add1000)	**0.228**	**0.654**	0.330	0.040	0.463
	Unimelb (5p)	0.198	0.623	0.374	0.056	0.475
	Unimelb (50k)	0.198	0.633	0.384	0.060	**0.494**
	UoS (#WN senses)	0.171	0.600	0.298	0.046	0.186
	UoS (top-3)	0.220	0.637	0.370	0.044	0.451
	La Sapienza (1)	0.131	0.544	0.332	–	–
	La Sapienza (2)	0.131	0.535	**0.394**	–	–
Sense embeddings	AdaGram, $\alpha = 0.05$, 100 dim. vectors	**0.274**	0.644	0.318	0.058	0.470
Single models	Cluster	0.196	0.652	0.319	0.032	0.610
	Depword	0.196	0.652	0.319	0.032	0.610
	Deptarget	0.189	0.655	0.314	0.025	0.610
	Depall	0.188	0.650	0.313	0.029	0.608
	Trigramtarget	0.179	0.632	0.303	0.009	**0.612**
	Trigramall	0.182	0.650	0.302	0.015	0.594
Feature comb.	DeptargetDepwordClusterTrigramtarget	0.188	0.654	0.317	0.032	**0.611**
	DepallDepwordClusterTrigramall	0.197	0.652	0.317	0.034	**0.611**
	DeptargetDepwordCluster	0.189	0.655	0.318	0.033	**0.611**
	DepallDepwordCluster	0.197	0.651	0.317	0.034	**0.611**
Meta comb.	Cluster+Deptarget+Depword+Trigramtarget: majority	0.197	0.645	0.317	0.037	0.600
	Cluster+Deptarget+Depword+Trigramtarget: ranks	**0.219**	**0.657**	0.309	0.034	0.487
	Cluster+Deptarget+Depword+Trigramtarget: sum	0.204	0.646	0.320	0.040	0.607
	Cluster+Depall+Depword+Trigramall: majority	0.196	0.646	0.315	0.035	0.601
	Cluster+Depall+Depword+Trigramall: ranks	0.216	0.654	0.316	0.042	0.526
	Cluster+Depall+Depword+Trigramall: sum	0.193	0.651	0.317	0.034	0.605
	Cluster+Deptarget+Depword: majority	0.200	0.647	0.317	0.039	0.601
	Cluster+Deptarget+Depword: ranks	**0.217**	**0.659**	**0.324**	**0.048**	0.533
	Cluster+Deptarget+Depword: sum	0.204	0.647	0.319	0.040	0.607
	Cluster+Depall+Depword: majority	0.200	0.647	0.317	0.039	0.601
	Cluster+Depall+Depword: ranks	0.200	0.646	0.317	0.039	0.601
	Cluster+Depall+Depword: sum	0.197	0.655	0.318	0.038	0.607

Table 3: Performance of our method on the nouns contexts from the SemEval 2013 Task 13 dataset. The models were trained on the ukWaC corpus. Top scores of the state-of-the-art systems (SemEval participants and the AdaGram) and of our systems are set in boldface; the best scores overall are underlined.

specific texts. The method takes as input a text corpus and learns an interpretable coarse-grained sense inventory, where each sense has a rich feature representation used for disambiguation.

The novel element of our approach is the use of an induced sense inventory as a pivot for aggregation and combination of heterogeneous context clues. This framework let us easily incorporate various context features in a single model. In our experiments we demonstrated combinations of four classes of features, but the framework can easily accommodate other types of features.

While other systems already used some features employed in our approach (e.g., the UoS system relies on dependency features), according to our knowledge, before there was no general methodology for incorporation of heterogenous features in an unsupervised WSD model.

The single-feature model based on dependency words proved to be most robust across tested datasets. As to the combination variants, we found it advantageous to combine all four types of features considered in our experiments. Combining models on the feature level yields highest F-scores in comparison to the meta-combinations. However, the meta-combination based on sum of confidences yields the most robust results across the datasets. Besides, the meta-combination based on sum of ranks provides higher precision at the cost of recall.

Experiments on a SemEval dataset, show that our approach performs comparably to the state-of-the-art unsupervised systems. Besides, the method perform almost no errors in the case of coarse-grained homonymous senses.

Implementation of our approach with several pre-trained models is available online.[5]

Acknowledgments

We acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) foundation under the project "JOIN-T: Joining Ontologies and Semantics Induced from Text".

[5] https://github.com/tudarmstadt-lt/JoSimText

References

Eneko Agirre and Philip 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.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of the AISTATS Conference*, Granada, 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, Georgia, USA.

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, USA.

Chris Biemann. 2010. Co-Occurrence Cluster Features for Lexical Substitutions in Context. In *Proceedings of the 5th Workshop on TextGraphs in conjunction with ACL*, pages 55–59, Uppsala, Sweden.

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.

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

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4): Can we beat Google*, pages 47–54.

David Hope and Bill Keller. 2013a. 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 - Volume Part I*, pages 368–381, Samos, Greece. Springer-Verlag.

David Hope and Bill Keller. 2013b. UoS: A Graph-Based System for Graded 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)*, number 1, pages 689–694, Atlanta, GA, USA.

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 ACL*, pages 873–882, Jeju Island, Korea.

David Jurgens and Ioannis Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. 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 290–299, Atlanta, Georgia, USA.

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-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, volume 8, pages 74–80, Philadelphia, PA, USA.

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, Georgia, USA.

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 ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, volume 10, pages 41–48, Philadelphia, PA.

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.

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.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of ICML*, volume 98, pages 296–304, Madison, WI, USA.

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.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A Semantic Concordance. In *Proceedings of the Workshop on Human Language Technology - HLT '93*, pages 303–308, NJ, USA.

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

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. 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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar.

Luis Nieto Piña and Richard Johansson. 2016. Embedding senses for efficient graph-based word sense disambiguation. In *Proceedings of TextGraphs-10, Proceedings of the Human Language Technology Conference of the NAACL*, pages 1–5, San Diego, USA.

Alexander Panchenko, Pavel Romanov, Olga Morozova, Hubert Naets, Andrey Philippovich, Alexey Romanov, and Cédrick Fairon. 2013. Serelex: Search and visualization of semantically related words. In *European Conference on Information Retrieval*, pages 837–840. Springer.

Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619. ACM.

Ted. Pedersen and Rebecca Bruce. 1997. Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 197–207, Providence, RI, USA.

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.

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

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

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

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 *COLING*, pages 151–160, Dublin, Ireland.

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*, pages 1–7, Taipei, Taiwan.

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.