Improving Hypernymy Extraction with Distributional Semantic Classes

Alexander Panchenko[1*], **Dmitry Ustalov**[2,3*], **Stefano Faralli**[3], **Simone P. Ponzetto**[3], **Chris Biemann**[1]
[*] these authors contributed equally
[1] University of Hamburg, Department of Informatics, Language Technology Group, Germany
[2] University of Mannheim, School of Business Informatics and Mathematics, Data and Web Science Group, Germany
[3] Ural Federal University, Institute of Natural Sciences and Mathematics, Russia
{panchenko,biemann}@informatik.uni-hamburg.de
{dmitry,stefano,simone}@informatik.uni-mannheim.de

Abstract

In this paper, we show how distributionally-induced semantic classes can be helpful for extracting hypernyms. We present methods for inducing sense-aware semantic classes using distributional semantics and using these induced semantic classes for filtering noisy hypernymy relations. Denoising of hypernyms is performed by labeling each semantic class with its hypernyms. On the one hand, this allows us to filter out wrong extractions using the global structure of distributionally similar senses. On the other hand, we infer missing hypernyms via label propagation to cluster terms. We conduct a large-scale crowdsourcing study showing that processing of automatically extracted hypernyms using our approach improves the quality of the hypernymy extraction in terms of both precision and recall. Furthermore, we show the utility of our method in the domain taxonomy induction task, achieving the state-of-the-art results on a SemEval'16 task on taxonomy induction.

Keywords: semantic classes, distributional semantics, hypernyms, co-hyponyms, word sense induction

1. Introduction

Hypernyms are useful in various applications, such as question answering (Zhou et al., 2013), query expansion (Gong et al., 2005), and semantic role labelling (Shi and Mihalcea, 2005) as they can help to overcome sparsity of statistical models. Hypernyms are also the building blocks for learning taxonomies from text (Bordea et al., 2016). Consider the following sentence: "This café serves fresh *mangosteen* juice". Here the infrequent word "mangosteen" may be poorly represented or even absent in the vocabulary of a statistical model, yet it can be substituted by lexical items with better representations, which carry close meaning, such as its hypernym "fruit" or one of its close co-hyponyms, e.g. "mango".

Currently available approaches to hypernymy extraction focus on the acquisition of individual binary hypernymy relations (Hearst, 1992; Snow et al., 2004; Weeds et al., 2014; Shwartz et al., 2016; Glavaš and Ponzetto, 2017). Frequencies of the extracted relations usually follow a power-law, with a long tail of noisy extractions containing rare words. We propose a method that performs post-processing of such noisy binary hypernyms using distributional semantics, cf. Figure 1. Namely, we use the observation that distributionally related words are often are co-hyponyms (Wandmacher, 2005; Heylen et al., 2008) and operationalize it to perform filtering of noisy relations by finding dense graphs composed of both hypernyms and co-hyponyms.

The contribution of the paper is an unsupervised method for post-processing of noisy hypernymy relations based on clustering of graphs of word senses induced from text. The idea to use distributional semantics to find hypernyms seems natural and has been widely used. However, the existing methods used distributional, yet *sense-unaware* and *local* features. We are the first to use *global sense-aware distributional structure* via the induced semantic classes to improve hypernymy extraction. The implementation of our method and the induced language resources (distributional semantic classes and cleansed hypernymy relations) are available online.[1]

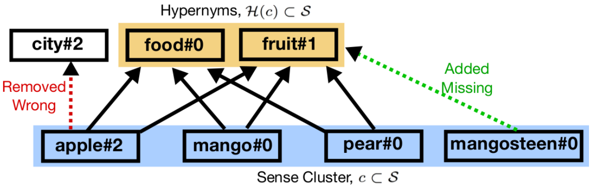

Hypernyms, $\mathcal{H}(c) \subset \mathcal{S}$
city#2
food#0
fruit#1
Removed
Wrong
Added
Missing
apple#2
mango#0
pear#0
mangosteen#0
Sense Cluster, $c \subset \mathcal{S}$

Figure 1: Our approach performs post-processing of hypernymy relations using distributionally induced semantic classes, represented by clusters of induced word senses labeled with noisy hypernyms. The word postfix, such as `#1`, is an ID of an induced sense. The wrong hypernyms outside the cluster labels are removed, while the missing ones not present in the noisy database of hypernyms are added.

2. Related Work

2.1. Extraction of Hypernyms

In her pioneering work, Hearst (1992) proposed to extract hypernyms based on lexical-syntactic patterns from text. Snow et al. (2004) learned such patterns automatically based on a set of hyponym-hypernym pairs. Pantel and Pennacchiotti (2006) presented another approach for weakly supervised extraction of similar extraction patterns. These approaches use some training pairs of hypernyms to bootstrap the pattern discovery process. For instance, Tjong Kim Sang (2007) used web snippets as a corpus for extraction of hypernyms. More recent approaches exploring the use of distributional word representations for extraction of

[1] `https://github.com/uhh-lt/mangosteen`

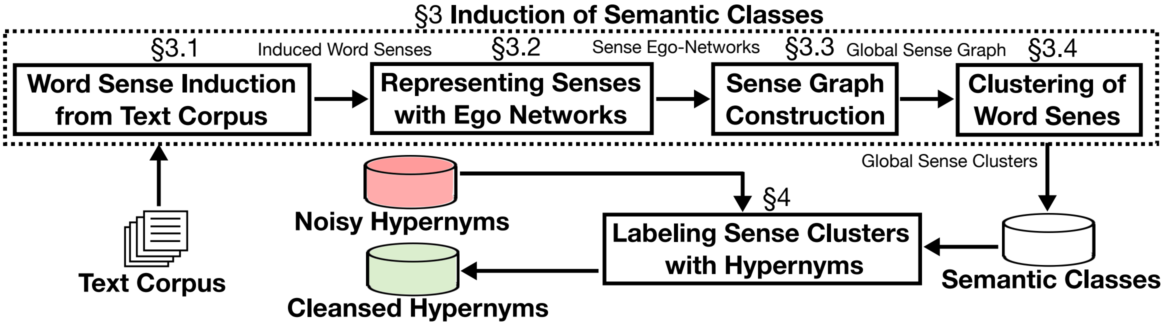


Figure 2: Outline of our approach: sense-aware distributional semantic classes are induced from a text corpus and then used to filter noisy hypernyms database (e.g. extracted by an external method from a text corpus).

hypernyms and co-hyponyms include (Roller et al., 2014; Weeds et al., 2014; Necsulescu et al., 2015; Vylomova et al., 2016). They rely on two distributional vectors to characterize a relation between two words, e.g. on the basis of the difference of such vectors or their concatenation. Levy et al. (2015) discovered a tendency to lexical memorization of such approaches, hampering their generalization to other domains.

Fu et al. (2014) relied on an alternative approach where a projection matrix is learned, which transforms a distributional vector of a hyponym to the vector of its hypernym. Ustalov et al. (2017a) improved this method by adding regularizers in the model that take into account negative training samples and the asymmetric nature of the hypernyms.

Recent approaches to hypernym extraction focused on learning *supervised* models based on a combination of syntactic patterns and distributional features (Shwartz et al., 2016). Note that while methods, such as (Mirkin et al., 2006) and (Shwartz et al., 2016) use distributional features for extraction of hypernyms, in contrast to our method, they do not take into account word senses and global distributional structure.

Seitner et al. (2016) performed extraction of hypernyms from the web-scale Common Crawl[2] text corpus to ensure high lexical coverage. In our experiments, we use this web-scale database of noisy hypernyms, as the large-scale repository of automatically extracted hypernyms to date.

2.2. Taxonomy and Ontology Learning

Most relevant in the context of automatic construction of lexical resource are methods for building resources from text (Caraballo, 1999; Biemann, 2005; Cimiano, 2006; Bordea et al., 2015; Velardi et al., 2013) as opposed to methods that automatically construct resources from semi-structured data (Auer et al., 2007; Navigli and Ponzetto, 2012) or using crowdsourcing (Biemann, 2013; Braslavski et al., 2016).

Our representation differs from the global hierarchy of words as constructed e.g. by (Berant et al., 2011; Faralli et al., 2016), as we are grouping many lexical items into a labeled sense cluster as opposed to organizing them in deep hierarchies. Kozareva and Hovy (2013) proposed a taxonomy induction method based on extraction of hypernyms using the doubly-anchored lexical patterns. Graph algorithms are used to induce a proper tree from the binary relations harvested from text.

2.3. Induction of Semantic Classes

This line of research starts with (Lin and Pantel, 2001), where sets of similar words are clustered into concepts. While this approach performs a hard clustering and does not label clusters, these drawbacks are addressed in (Pantel and Lin, 2002), where words can belong to several clusters, thus representing senses, and in (Pantel and Ravichandran, 2004), where authors aggregate hypernyms per cluster, which come from Hearst patterns. The main difference to our approach is that we explicitly represent senses both in clusters and in their hypernym labels, which enables us to connect our sense clusters into a global taxonomic structure. Consequently, we are the first to use semantic classes to improve hypernymy extraction.

Ustalov et al. (2017b) proposed a synset induction approach based on global clustering of word senses. The authors used the graph constructed of dictionary synonyms, while we use distributionally-induced graphs of senses.

3. Unsupervised Induction of Distributional Sense-Aware Semantic Classes

As illustrated in Figure 2, our method induces a sense inventory from a text corpus using the method of (Faralli et al., 2016; Biemann et al., 2018), and clusters these senses. Sample word senses from the induced sense inventory are presented in Table 1. The difference of the induced sense inventory from the sense clustering presented in Table 2 is that word senses in the induced resource are specific to a given target word, e.g. words “apple” and “mango” have distinct “fruit” senses, represented by a list of related senses. On the other hand, sense clusters represent a global and not a local clustering of senses, i.e. the “apple” in the “fruit” sense can be a member of only one cluster. This is similar to WordNet, where one sense can only belong to a single synset. Below we describe each step of our method.

3.1. Word Sense Induction from a Text Corpus

Each word sense s in the induced sense inventory $\mathcal{S}$ is represented by a list of neighbors $\mathcal{N}(s)$, see Table 1 for an example. Extraction of this network is performed using the method of Faralli et al. (2016) and involves three steps: (1) building a distributional thesaurus, i.e. a graph

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

ID: Word Sense, $s \in \mathcal{S}$	**Local Sense Cluster: Related Senses,** $\mathcal{N}(s) \subset \mathcal{S}$	**Hypernyms,** $\mathcal{H}(s) \subset \mathcal{S}$
mango#0	peach#1, grape#0, plum#0, apple#0, apricot#0, watermelon#1, banana#1, coconut#0, pear#0, fig#0, melon#0, **mangosteen#0**, ...	fruit#0, food#0, ...
apple#0	mango#0, pineapple#0, banana#1, melon#0, grape#0, peach#1, watermelon#1, apricot#0, cranberry#0, pumpkin#0, **mangosteen#0**, ...	fruit#0, crop#0, ...
Java#1	C#4, Python#3, Apache#3, Ruby#6, Flash#1, C++#0, SQL#0, ASP#2, Visual Basic#1, CSS#0, Delphi#2, MySQL#0, Excel#0, Pascal#0, ...	programming language#3, language#0, ...
Python#3	PHP#0, Pascal#0, Java#1, SQL#0, Visual Basic#1, C++#0, JavaScript#0, Apache#3, Haskell#5, .NET#1, C#4, SQL Server#0, ...	language#0, technology#0, ...

Table 1: Sample induced sense inventory entries representing “fruits” and “programming language” senses. Each word sense s is represented with a list of related senses $\mathcal{N}(s)$ and the list of hypernyms $\mathcal{H}(s)$. The hypernyms can be used as human-interpretable sense labels of the sense clusters. One sense s, such as “apple#0”, can appear in multiple entries.

ID	**Global Sense Cluster: Semantic Class,** $c \subset \mathcal{S}$	**Hypernyms,** $\mathcal{H}(c) \subset \mathcal{S}$
1	peach#1, banana#1, pineapple#0, berry#0, blackberry#0, grapefruit#0, strawberry#0, blueberry#0, fruit#0, grape#0, melon#0, orange#0, pear#0, plum#0, raspberry#0, watermelon#0, apple#0, apricot#0, watermelon#0, pumpkin#0, berry#0, **mangosteen#0**, ...	vegetable#0, fruit#0, crop#0, ingredient#0, food#0, ·
2	C#4, Basic#2, Haskell#5, Flash#1, Java#1, Pascal#0, Ruby#6, PHP#0, Ada#1, Oracle#3, Python#3, Apache#3, Visual Basic#1, ASP#2, Delphi#2, SQL Server#0, CSS#0, AJAX#0, JavaScript#0, SQL Server#0, Apache#3, Delphi#2, Haskell#5, .NET#1, CSS#0, ...	programming language#3, technology#0, language#0, format#2, app#0

Table 2: Sample of the induced sense clusters representing “fruits” and “programming language” semantic classes. Similarly to the induced word senses, the semantic classes are labeled with hypernyms. In contrast to the induced word senses, which represent a local clustering of word senses (related to a given word) semantic classes represent a global sense clustering of word senses. One sense c, such as “apple#0”, can appear only in a single cluster.

of related ambiguous terms (Biemann and Riedl, 2013); (2) word sense induction via clustering of ego networks (Widdows and Dorow, 2002; Everett and Borgatti, 2005) of related words using the Chinese Whispers graph clustering algorithm (Biemann, 2006); (3) disambiguation of related words and hypernyms. The word sense inventory used in our experiment[3] was extracted from a 9.3 billion tokens corpus, which is a concatenation of Wikipedia[4], ukWac (Ferraresi et al., 2008), LCC (Richter et al., 2006) and Gigaword (Graff and Cieri, 2003). Note that analogous graphs of senses can be obtained using word sense embeddings, see (Neelakantan et al., 2014; Bartunov et al., 2016). Similarly to any other distributional word graph, the induced sense inventory sense network is scale-free, cf. (Steyvers and Tenenbaum, 2005). Our experiments show that a global clustering of this network can lead to a discovery of giant components, which are useless in our context as they represent no semantic class. To overcome this problem, we re-build the sense network as described below.

3.2. Representing Senses with Ego Networks

To perform a global clustering of senses, we represent each induced sense s by a second-order *ego network* (Everett and Borgatti, 2005). An ego network is a graph consisting of all related senses $\mathcal{R}(s)$ of the ego sense s reachable via a path of length one or two, defined as:

$$\{s_j : (s_j \in \mathcal{N}(s)) \vee (s_i \in \mathcal{N}(s) \wedge s_j \in \mathcal{N}(s_i))\}. \quad (1)$$

Each edge weight $\mathcal{W}_s(s_i, s_j)$ between two senses is taken from the induced sense inventory network (Faralli et al., 2016) and is equal to a distributional semantic relatedness score between s_i and s_j.

Senses in the induced sense inventory may contain a mixture of different senses introducing noise in a global clustering: cf. Figure 3, where “Python” in the animal sense is related to both car and snake senses. To minimize the impact of the word sense induction errors, we filter out ego networks with a highly segmented structure. Namely, we cluster each ego network with the Chinese Whispers algorithm and discard networks for which the cluster containing the target sense s contains less than 80% nodes of the respective network to ensure semantic coherence inside the word groups. Besides, all nodes of a network not appearing in the cluster containing the ego sense s are also discarded.

3.3. Global Sense Graph Construction

The goal of this step is to merge ego networks of individual senses constructed at the previous step into a global graph. We compute weights of the edges of the global graph by counting the number of co-occurrences of the same edge in different networks:

$$\mathcal{W}(s_i, s_j) = \sum_{s \in \mathcal{S}} \mathcal{W}_s(s_i, s_j). \quad (2)$$

For filtering out noisy edges, we remove all edges with the weight less than a threshold t. Finally, we apply the function $E(w)$ that re-scales edge weights. We tested identity function (count) and the natural logarithm (log):

$$\mathcal{W}(s_i, s_j) = \begin{cases} E(\mathcal{W}(s_i, s_j)) & \text{if } \mathcal{W}(s_i, s_j) \geq T, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

[3] The input and output datasets are available for download at https://doi.org/10.5281/zenodo.1174041

[4] https://doi.org/10.5281/zenodo.229904

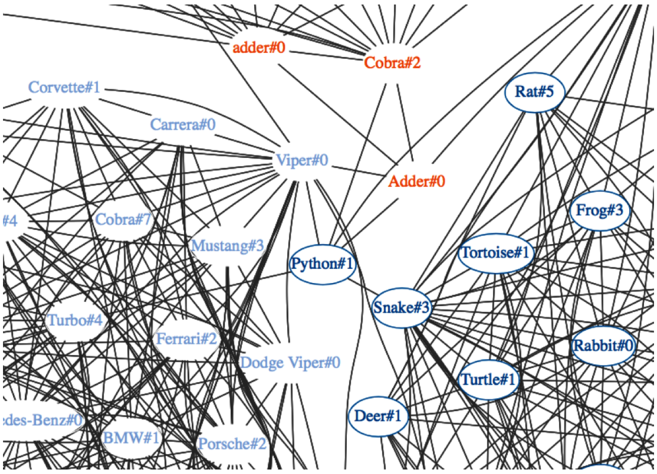


Figure 3: An example of a non-coherent ego network of the automatically induced sense `Python#1`, representing the "animal" sense. We prune it to remove terms not relevant to the animal sense.

3.4. Clustering of Word Senses

The core of our method is the induction of semantic classes by clustering the global graph of word senses. We use the Chinese Whispers algorithm to make every sense appear only in one cluster c. Results of the algorithm are groups of strongly related word senses that represent different concepts (cf. Figure 4). Hypernymy is by definition a relation between nouns. Thus optionally, we remove all single-word senses that do not correspond to nouns using the Pattern library (De Smedt and Daelemans, 2012). This optional mode is configured by the boolean parameter N.

We use two clustering versions in our experiments: the *fine-grained* model clusters 208,871 induced word senses into 1,870 semantic classes, and the *coarse-grained* model that groups 18,028 word senses into 734 semantic classes. To find optimal parameters of our method, we compare the induced labeled sense clusters to lexical semantic knowledge from WordNet 3.1 (Fellbaum, 1998) and BabelNet 3.7 (Navigli and Ponzetto, 2012).

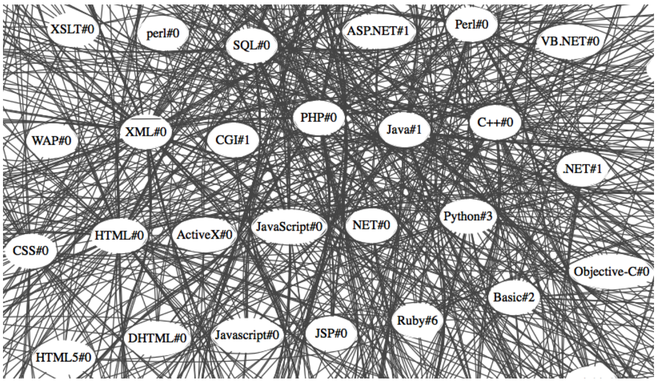


Figure 4: Senses referring to programming languages co-occur in global sense cluster entries, resulting in a densely connected set of co-hyponyms.

4. Denoising Hypernyms using the Induced Distributional Semantic Classes

By labeling the induced semantic classes with hypernyms we can thereby remove wrong ones or add those that are missing as illustrated in Figure 1. Each sense cluster is labeled with the noisy input hypernyms, where the labels are the common hypernyms of the cluster word (cf. Table 2). Hypernyms that label no sense cluster are filtered out. In addition, new hypernyms can be generated as a result of labeling. Additional hypernyms are discovered by propagating cluster labels to the rare words without hypernyms, e.g. "mangosteen" in Figure 1. For labeling we used the tf-idf weighting. Hypernyms that appear in many senses s are weighted down:

$$\text{tf-idf}(h) = \sum_{s \in c} \mathcal{H}(s) \cdot \log \frac{|\mathcal{S}|}{|h \in \mathcal{H}(s) : \forall s \in \mathcal{S}|}, \quad (4)$$

where $\sum_{s \in c} \mathcal{H}(s)$ is a sum of weights for all hypernyms for each sense s, per each cluster c.

We label each sense cluster c with its top five hypernyms $\mathcal{H}(c)$. Each hypernym is disambiguated using the method of Faralli et al. (2016). Namely, we calculate the cosine similarity between the context (the current sense cluster) and the induced senses (local clusters of the ambiguous word).

Distributional representations of rare words, such as "mangosteen" can be less precise than those of frequent words. However, co-occurrence of a hyponym and a hypernym in a single sentence is not required in our approach, while it is the case for the path-based hypernymy extraction methods.

5. Finding an Optimal Configuration of Meta Parameters of the Method

The approach consists of several sequential stages, as depicted in Figure 2, with each stage having a few meta parameters. This study is designed to find promising combinations of these meta parameters. In this section, we propose several metrics which aim at finding an optimal configuration of all these meta parameters jointly. In particular, to compare different configurations of our approach, we compare the labeled sense clusters to WordNet 3.1 (Fellbaum, 1998) and BabelNet 3.7 (Navigli and Ponzetto, 2012). The assumption is that the optimal model contains lexical semantic knowledge similar to the knowledge in the lexical resources. To implement the evaluation metrics we used the NLTK library (Bird et al., 2009) and the BabelNet Java API.[5]

5.1. Metrics Quantifying Goodness of Fit of the Induced Structures to Lexical Resources

To summarize various aspects of the lexical resource, we propose a score that is maximized if labeled sense clusters are generated directly from a lexical resource:

$$hpc\text{-score}(c) = \frac{h\text{-score}(c) + 1}{p\text{-score}(c) + 1} \cdot \text{coverage}(c). \quad (5)$$

p-score(c) quantifies the plausibility of the sense cluster c.

[5] `http://www.babelnet.org`

	Min. sense co-occurrences, t	Edge weight, E	Only nouns, N	Hypernym weight, H	Number of clusters	Number of senses	*hpc*-avg, WordNet	*hpc*-avg, BabelNet
coarse-gr.	100	log	yes	tf-idf	734	18 028	**0.092**	**0.304**
	100	log	no	tf-idf	763	27 149	0.090	0.303
	100	count	no	tf-idf	765	27 149	0.089	0.302
	100	log	no	tf	784	27 149	0.090	0.300
	100	count	yes	tf	733	18 028	**0.092**	0.299
	100	count	no	tf	772	27 149	0.089	0.297
	100	count	yes	tf-idf	732	18 028	0.091	0.295
	100	log	yes	tf	726	18 028	0.088	0.293
fine-gr.	0	count	no	tf-idf	1870	208 871	**0.041**	**0.279**
	0	count	no	tf	1877	208 871	**0.041**	0.278
	0	count	yes	tf	2070	144 336	0.037	0.240
	0	count	yes	tf-idf	2080	144 336	0.038	0.240
	0	log	yes	tf-idf	4709	144 336	0.027	0.138
	0	log	yes	tf	4679	144 336	0.027	0.136
	0	log	no	tf-idf	5960	208 871	0.035	0.127
	0	log	no	tf	5905	208 871	0.036	0.126

Table 3: Performance of different configurations of the hypernymy labeled global sense clusters in terms of their similarity to WordNet/BabelNet. The results are sorted by performance on BabelNet dataset, the best values in each section are boldfaced. The two underlined configurations are respectively the best *coarse-grained* and *fine-grained* grained semantic class models used in all experiments. The coarse grained model contains less semantic classes, but they tend to be more consistent than those of the fine-grained model, which contains more senses and classes.

It reflects the distance of co-hyponyms in a lexical resource:

$$p\text{-score}(c) = \frac{1}{|c|} \sum_{i=1}^{|c|} \sum_{j=1}^{i} \text{dist}(w_i, w_j). \quad (6)$$

The lower the p-score is, the closer the hyponyms are located in the gold standard resource. For each pair of distinct lemmas (w_i, w_j) in the cluster of co-hyponyms c, we search for the minimal shortest path distance (SPD) between the synsets corresponding to each word in the pair, i.e. $S(w_i)$ is the set of synsets having the w_i lemma and $S(w_j)$ is the similar set with respect to the w_j lemma:

$$\text{dist}(w_i, w_j) = \min_{\substack{s' \in S(w_i),\\ s'' \in S(w_j)}} \text{SPD}(s', s''). \quad (7)$$

h-score(c) quantifies plausibility of the hypernyms $\mathcal{H}(c)$ of a sense cluster c measuring the precision of extracted hypernyms:

$$h\text{-score}(c) = \frac{|\mathcal{H}(c) \cap \text{gold}(c)|}{|\mathcal{H}(c)|}. \quad (8)$$

The gold(c) is composed of the lowest common hypernyms (LCH) in the lexical resource for each pair of lemmas in the sense cluster c:

$$\text{gold}(c) = \bigcup_{\substack{w_i \in c,\\ w_j \in c}} \bigcup_{\substack{s' \in S(w_i),\\ s'' \in S(w_j)}} \{\text{LCH}(s', s'')\}. \quad (9)$$

coverage(c) quantifies how well cluster words are represented in the gold standard resource. Thus, errors in poorly represented clusters are discounted via coverage. Coverage is the fraction of the lemmas appearing both in the cluster c and in the vocabulary of the resource $\mathcal{V}$:

$$\text{coverage}(c) = \frac{|c \cap \mathcal{V}|}{|c|}. \quad (10)$$

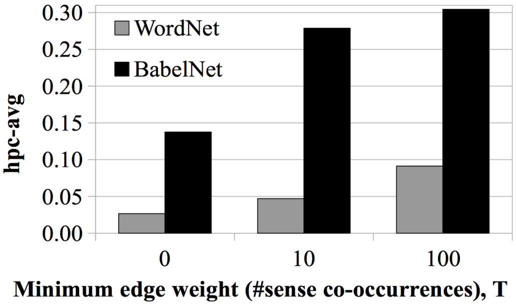


Figure 5: Impact of the min. edge weight t.

The total score used to rank various configurations of our approach averages hpc-score scores for all induced sense clusters:

$$hpc\text{-avg} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} hpc\text{-score}(c). \quad (11)$$

5.2. Results

Meta parameter search results based on the comparison to WordNet and BabelNet are provided in Figure 5 and Table 3. The minimal edge weight t trades off between the size of the resulting resource (number of words and senses) and its similarity to the gold lexical resources. The higher the threshold, the fewer nodes remain in the graph, yet these remaining nodes form densely interlinked communities. For t of 100, each pair of senses in the graph is observed in at least 100 ego networks. Secondly, for the unpruned model ($t = 0$), edge weights based on counts worked better than logarithmic weights. However, when pruned ($t > 0$), logarithmic edge weighting shows better results. Thirdly, the tf-idf weights proved to yield consistent improvements over the basic tf weighting. For the pruned model, the variation in scores across different configurations is small as the underlying graphs are of high quality, while for the unpruned model the choice of parameters has much more impact as the sense graphs are noisier.

We selected the best-performed configuration according to BabelNet (hpc-avg of 0.304), which also is the second best configuration according to WordNet (hpc-avg of 0.092). This model is based on the edge threshold t of 100, logarithmic weights of edges contains only nouns and hypernyms ranked according to $\mathrm{tf\text{-}idf}$. Note also that, the best-unpruned model ($t = 0$) has BabelNet hpc-avg of 0.279, which is only 10% lower than the best model, yet the unpruned model has an order of magnitude larger vocabulary and a more fine-grained representation (734 vs. 1,870 clusters). Thus, if coverage is important, the unpruned model is recommended. In the remainder of this paper, we continue with the first model listed in Table 3 and evaluate it in the following experiments.

6. Evaluation

To evaluate our approach, we conduct two intrinsic evaluations and one extrinsic evaluation. The first experiment aims to estimate the fraction of spurious sense clusters, the second one evaluates the quality of the post-processed hypernyms. Finally, we evaluate the induced semantic classes in application to the taxonomy induction task.

6.1. Experiment 1: Plausibility of the Induced Semantic Classes

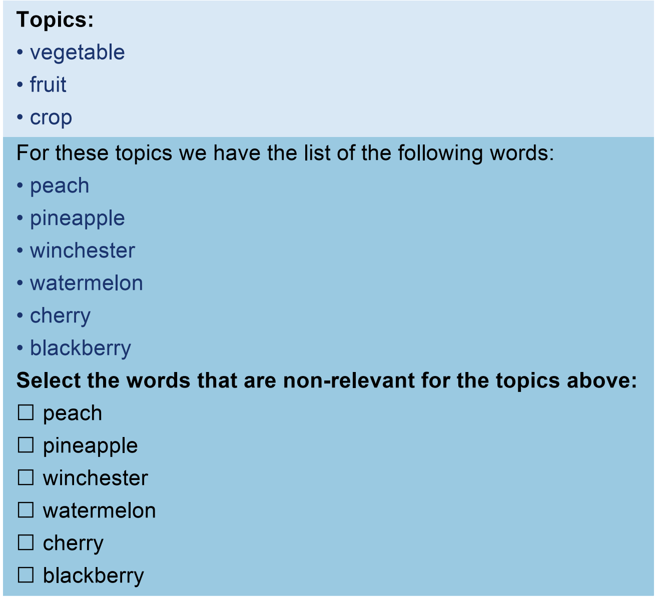
Comparison to gold standard resources allows us to gauge the relative performances of various configurations of our method. To measure the absolute quality of the best configuration selected in the previous section, we rely on microtask-based crowdsourcing with CrowdFlower[6].

6.1.1. Task Design

We used two crowdsourcing tasks based on word intruder detection (Chang et al., 2009) to measure how humans perceive the extracted lexical-semantic structures. Namely, the tasks are designed to evaluate the quality of the extracted sense clusters and their labels. The input form presented to an annotator is illustrated in Figure 6. A crowdworker is asked to identify words that do not match the context represented by words from a sense cluster or its label. To generate an intruder, following the original design of Chang et al. (2009), we select a random word from a cluster and replace it with a word of similar frequency that does not belong to any cluster (bias here is low as the evaluated model contains 27,149 out of 313,841 induced word senses). In both tasks, the workers have been provided with concise instructions and test questions.

6.1.2. Evaluation Metrics

We compute two metrics on the basis on annotation results: (1) *accuracy* is the fraction of tasks where annotators correctly identified the intruder, thus the words from the cluster are consistent; (2) *badness* is the fraction of tasks for which non-intruder words were selected. In this experiment, we assume that it is easy to identify the intruder in a correct sense cluster and difficult in a noisy, implausible sense cluster. We compute *accuracy* as the fraction of tasks where annotators correctly identified the intruder, thus the words from the cluster are consistent.



Topics:
- vegetable
- fruit
- crop

For these topics we have the list of the following words:
- peach
- pineapple
- winchester
- watermelon
- cherry
- blackberry

Select the words that are non-relevant for the topics above:

□ peach
□ pineapple
□ winchester
□ watermelon
□ cherry
□ blackberry

Figure 6: Layout of the sense cluster evaluation crowdsourcing task, the entry "winchester" is the intruder.

	Accuracy	**Badness**	**Randolph κ**
Sense clusters, c	0.859	0.248	0.739
Hyper. labels, $\mathcal{H}(c)$	0.919	0.208	0.705

Table 4: Plausibility of the sense clusters according to human judgments via an intruder detection experiment for the coarse-grained semantic class model.

6.1.3. Results

Table 4 summarizes the results of the intruder detection experiment. Overall, 68 annotators provided 2,035 judgments about the quality of sense clusters. Regarding hypernyms, 98 annotators provided 2,245 judgments. The majority of the induced semantic classes and their labels are highly plausible according to human judgments: the accuracy of the sense clusters based on the intruder detection is 0.859 (agreement of 87%), while the accuracy of hypernyms is 0.919 (agreement of 85%). The Randolph κ of respectively 0.739 and 0.705 indicates substantial inter-observer agreement (Randolph, 2005).

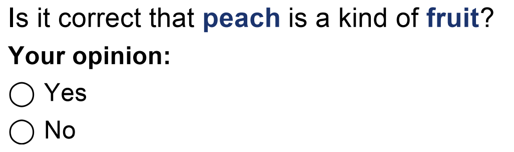
According to the feedback mechanism of the CrowdFlower, the co-hyponymy task received a 4.0 out of 5.0 rating, while the hypernymy task received a 4.4 out of 5.0 rating. The crowdworkers show a *substantial* agreement according to Randolph κ coefficient computed 0.739 for the cluster evaluation task and 0.705 for the hypernym evaluation task.

Major sources of errors for crowdworkers are rare words and entities. While clusters with well-known entities, such as "Richard Nixon" and "Windows Vista" are correctly labeled, examples of other less-known named entities, e.g. cricket players, are sometimes wrongly labeled as implausible. Another source of errors during crowdsourcing were wrongly assigned hypernyms: in rare cases, sense clusters are labeled with hypernyms like "thing" or "object" that are

[6] https://www.crowdflower.com

	Precision	Recall	F-score
Original hypernymy relations extracted from the Common Crawl corpus (Seitner et al., 2016)	0.475	0.546	0.508
Enhanced hypernyms with the *coarse-grained* semantic classes	**0.541**	**0.679**	**0.602**

Table 5: Results of post-processing of a noisy hypernymy database with our approach, evaluated using human judgements.


Is it correct that **peach** is a kind of **fruit**?

Your opinion:

○ Yes

○ No

Figure 7: Layout of the hypernymy annotation task.

too generic even under tf-idf weighting.

6.2. Experiment 2: Improving Binary Hypernymy Relations

In this experiment, we test whether our post-processing based on the semantic class improves the quality of hypernymy relations (cf. Figure 2).

6.2.1. Generation of Binary Hypernyms.

We evaluated the best coarse-grained model identified in the first experiment (t of 100). Each sense cluster of this model is split into the set $H_{cluster}$ of binary hypernyms, as illustrated in Figure 1. Overall, we gathered 85,290 hypernym relations for 17,058 unique hyponyms. Next, we gathered the set H_{orig} of 75,486 original hypernyms for exactly the same 17,058 hyponyms. For each word from the sense cluster we looked up top five hypernyms under the best ones when sorting them by extraction frequency from the hypernym relation database of Seitner et al. (2016) as in our model each sense cluster is labeled with five hypernyms from the same database. The database of Seitner et al. (2016) is extracted using lexical patterns. Note that any other method for extraction of binary hypernyms can be used at this point, e.g. (Weeds et al., 2014; Roller et al., 2014; Shwartz et al., 2016; Glavaš and Ponzetto, 2017). For the comparison, we gathered up to five hypernyms for each word, using (1) the most frequent hypernym relations from (Seitner et al., 2016) vs. (2) the cluster labeling method as described above.

6.2.2. Task Design

We drew a random sample of 4,870 relations using lexical split by hyponyms. All relations from $H_{cluster}$ and H_{orig} of one hyponym were included in the sample. These relations were subsequently annotated by human judges using crowdsourcing. We asked crowdworkers to provide a binary judgment about the correctness of each hypernymy relation as illustrated in Figure 7.

6.2.3. Results

Overall, 298 annotators completed 4,870 unique tasks each labeled 6.9 times on average, resulting in a total of 33,719 binary human judgments about hypernyms. We obtained a *fair* agreement among annotators of 0.548 in terms of the Randolph κ (Meyer et al., 2014). Since CrowdFlower reports a confidence for each answer, we selected $N = 3$ most confident answers per pair and aggregated them using weighted majority voting. The ties were broken pessimistically, i.e. by treating a hypernym as irrelevant. Results for $N \in 3, 5, 6$ varied less than by 0.002 in terms of F-score. The task received the rating of a 4.4 out of 5.0 according to the annotator's feedback mechanism.

Table 5 presents results of the experiment. Since each pair received a binary score, we calculated Precision, Recall, and F-measure of two compared methods. Our denoising method improves the quality of the original hypernyms by a large margin both in terms of precision and recall, leading to an overall improvement of 10 F-score points. The improvements of recall are due to the fact that to label a cluster of co-hyponyms it is sufficient to lookup hypernyms for only a fraction of words in the clusters. However, binary relations will be generated between all cluster hypernyms and the cluster words potentially generating hypernyms missing in the input database. For instance, a cluster of fruits can contain common entries like "apple" and "mango" which ensure labeling it with the word "fruit". Rare words in the same cluster, like "mangosteen", which have no hypernyms in the original resource due to the sparsity of the pattern-based approach, will also obtain the hypernym "fruit" as they are distributionally related to frequent words with reliable hypernym relations, cf. Figure 1. We also observed this effect frequently with clusters of named entities, like cricket players. Improvements in precision are due to filtering of wrong extractions, which are different for different words and thus top hypernyms of a cluster contain only hypernyms confirmed by several co-hyponyms.

Finally, note that all previous hypernymy extraction methods output binary relations between undisambiguated words (cf. Section 2.). Therefore, our approach could be used to improve results of other state-of-the-art hypernymy extraction approaches, such as HypeNET (Shwartz et al., 2016).

6.3. Experiment 3: Improving Domain Taxonomy Induction

In this section, we show how the labeled semantic classes can be used for induction of domain taxonomies.

6.3.1. SemEval 2016 Task 13

We use the taxonomy extraction evaluation dataset by Bordea et al. (2016), featuring gold standard taxonomies for three domains (Food, Science, Environment) and four languages (English, Dutch, French, and Italian) on the basis of existing lexical resources, such as WordNet and Eurovoc (Steinberger et al., 2006).[7] Participants were supposed to build a taxonomy provided a vocabulary of a domain. Since our other experiments were conducted on English, we used the English part of the task. The evaluation is

[7] http://eurovoc.europa.eu

System / Domain, Dataset	Food, WordNet	Science, WordNet	Food, Combined	Science, Combined	Science, Eurovoc	Environment, Eurovoc
WordNet	1.0000	1.0000	0.5870	0.5760	0.6243	n.a.
Baseline	0.0022	0.0016	0.0019	0.0163	0.0056	0.0000
JUNLP	0.1925	0.0494	0.2608	0.1774	0.1373	0.0814
NUIG-UNLP	n.a.	0.0027	n.a.	0.0090	0.1517	0.0007
QASSIT	n.a.	0.2255	n.a.	0.5757	0.3893	0.4349
TAXI	0.3260	0.2255	0.2021	0.3634	0.3893	0.2384
USAAR	0.0021	0.0008	0.0000	0.0020	0.0023	0.0007
Semantic Classes (fine-grained)	0.4540	0.4181	0.5147	0.6359	**0.5831**	0.5600
Semantic Classes (coarse-grained)	**0.4774**	**0.5927**	**0.5799**	**0.6539**	0.5515	**0.6326**

Table 6: Comparison of the our taxonomy induction method on the SemEval 2016 Task 13 on Taxonomy Extraction Evaluation (Bordea et al., 2016) for English in terms of cumulative Fowlkes&Mallows measure (F&M).

Domain	#Seeds words	#Expand. words	#Clusters, fine-gr.	#Clusters, coarse-gr.
Food	2 834	3 047	29	21
Science	806	1 137	73	35
Environ.	261	909	111	39

Table 7: Summary of the domain-specific sense clusters.

based on the Fowlkes&Mallows Measure (F&M), a cumulative measure of the similarity of both taxonomies (Velardi et al., 2013).

6.3.2. Taxonomy Induction using Semantic Classes

Our method for taxonomy induction takes as input a vocabulary of the domain and outputs a taxonomy of the domain. The method consists of three steps: (1) retrieving sense clusters relevant to the target domain; (2) generation of binary relations though a Cartesian product of words in a sense cluster and its labels; (3) attaching disconnected components to the root (the name of the domain). We retrieve domain-specific senses for each domain of the SemEval datasets by a lexical filtering. First, we build an extended lexicon of each domain on the basis of the seed vocabulary of the domain provided in the SemEval dataset. Namely, for each seed term, we retrieve all semantically similar terms. To filter out noisy expansions, related terms are added to the expanded vocabulary only if there are at least $k = 5$ common terms between the seed vocabulary and the list of related terms. Second, we retrieve all sense clusters that contain at least one term from the expanded vocabulary among its sense clusters or hypernyms. Table 7 summarizes results of this domain filtering. After, we generate binary hypernymy relations by linking every word in the semantic class to each hypernymy label as shown in Figure 1. Finally, we link roots of each disconnected components to the root of the taxonomy, e.g. "food" for the Food domain. Note that this step was used by SemEval participants, e.g. in the TAXI system (Panchenko et al., 2016).

6.3.3. Results

Table 6 presents results of the taxonomy extraction experiment. We evaluated two best models of our method: a *coarse* and a *fine* grained clusterings featuring respectively 734 and 1870 semantic classes identified in Section 5. with different levels of pruning: $t \in \{0, 100\}$. As one can observe, our model based on the labeled sense clusters significantly outperforms the substring-based baseline and all participating system by a large margin on all domains. For the "Science (Eurovoc)" and "Food" domains our method yields results comparable to WordNet while remaining unsupervised and knowledge-free. Besides, for the "Science" domain our method outperforms WordNet, indicating on the high quality of the extracted lexical semantic knowledge. Overall, the *coarse-grained* more pruned model yielded better results as compared to *fine-grained* un-pruned model for all domains but "Science (Eurovoc)".

7. Conclusion

In this paper, we presented an unsupervised method for the induction of sense-aware semantic classes using distributional semantics and graph clustering and showed how these can be used for post-processing of noisy hypernymy databases extracted from text. We determined optimal parameters of our approach by a comparison to existing lexical-semantic networks. To evaluate our approach, we performed three experiments. A large-scale crowdsourcing study indicated a high plausibility of extracted semantic classes according to human judgment. Besides, we demonstrated that our approach helps to improve precision and recall of a hypernymy extraction method. Finally, we showed how the proposed semantic classes can be used to improve domain taxonomy induction from text.

While we have demonstrated the utility of our approach for hypernym extraction and taxonomy induction, we believe that the induced semantic classes can be useful in other tasks. For instance, in (Panchenko et al., 2017) these semantic classes were used as an inventory for word sense disambiguation to deal with out of vocabulary words.

8. Acknowledgements

This research was supported by Deutscher Akademischer Austauschdienst (DAAD), Deutsche Forschungsgemeinschaft (DFG) under the project "Joining Ontologies and Semantics Induced from Text" (JOIN-T), and by the Ministry of Education and Science of the Russian Federation Agreement no. 02.A03.21.0006. We are grateful to three anonymous reviewers for their helpful comments. Finally, we are grateful to Dirk Johannßen for providing feedback on an early version of this paper.

9. Bibliographical References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007: The Semantic Web*, pages 722–735. Springer Berlin Heidelberg, Busan, Korea.

Bartunov, S., Kondrashkin, D., Osokin, A., and Vetrov, D. (2016). Breaking sticks and ambiguities with adaptive skip-gram. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 130–138, Cadiz, Spain.

Berant, J., Dagan, I., and Goldberger, J. (2011). Global Learning of Typed Entailment Rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA. Association for Computational Linguistics.

Biemann, C. and Riedl, M. (2013). Text: now in 2D! A framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

Biemann, C., Faralli, S., Panchenko, A., and Ponzetto, S. P. (2018). A framework for enriching lexical semantic resources with distributional semantics. *Natural Language Engineering*, pages 1–48.

Biemann, C. (2005). Ontology Learning from Text: A Survey of Methods. *GLDV-Journal for Computational Linguistics and Language Technology*, 20(2):75–93.

Biemann, C. (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*, TextGraphs-1, pages 73–80, New York, NY, USA. Association for Computational Linguistics.

Biemann, C. (2013). Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, 47(1):97–122.

Bird, S., Loper, E., and Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

Bordea, G., Buitelaar, P., Faralli, S., and Navigli, R. (2015). SemEval-2015 Task 17: Taxonomy Extraction Evaluation (TExEval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910, Denver, CO, USA, June. Association for Computational Linguistics.

Bordea, G., Lefever, E., and Buitelaar, P. (2016). SemEval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *SemEval-2016*, pages 1081–1091, San Diego, CA, USA. Association for Computational Linguistics.

Braslavski, P., Ustalov, D., Mukhin, M., and Kiselev, Y. (2016). YARN: Spinning-in-Progress. In *Proceedings of the 8th Global WordNet Conference*, GWC 2016, pages 58–65, Bucharest, Romania. Global WordNet Association.

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 120–126, College Park, MD, USA. Association for Computational Linguistics.

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., and Blei, D. M. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc.

Cimiano, P., (2006). *Ontology Learning from Text*, pages 19–34. Springer US, Boston, MA, USA.

De Smedt, T. and Daelemans, W. (2012). Pattern for Python. *Journal of Machine Learning Research*, 13:2031–2035.

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

Faralli, S., Panchenko, A., Biemann, C., and Ponzetto, S. P. (2016). Linked Disambiguated Distributional Semantic Networks. In *The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Part II*, Lecture Notes in Computer Science, pages 56–64, Kobe, Japan. Springer International Publishing.

Fellbaum, C. (1998). *WordNet: An Electronic Database*. MIT Press.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (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, Marrakech, Morocco. European Language Resources Association (ELRA).

Fu, R., Guo, J., Qin, B., Che, W., Wang, H., and Liu, T. (2014). Learning Semantic Hierarchies via Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, MD, USA. Association for Computational Linguistics.

Glavaš, G. and Ponzetto, S. P. (2017). Dual tensor model for detecting asymmetric lexico-semantic relations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1758–1768, Copenhagen, Denmark, September. Association for Computational Linguistics.

Gong, Z., Cheang, C. W., and Leong Hou, U. (2005). Web Query Expansion by WordNet. In *Proceedings of the 16th International Conference on Database and Expert Systems Applications - DEXA '05*, pages 166–175, Copenhagen, Denmark. Springer Berlin Heidelberg.

Graff, D. and Cieri, C. (2003). English Gigaword corpus. *Linguistic Data Consortium*.

Hearst, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING '92, pages 539–545, Nantes, France. Association for Computational Linguistics.

Heylen, K., Peirsman, Y., Geeraerts, D., and Speelman, D. (2008). Modelling Word Similarity: an Evaluation of Automatic Synonymy Extraction Algorithms. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, LREC 2008, pages

3243–3249, Marrakech, Morocco. European Language Resources Association (ELRA).

Kozareva, Z. and Hovy, E. (2013). Tailoring the automated construction of large-scale taxonomies using the web. *Language resources and evaluation*, 47(3):859–890.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do Supervised Distributional Methods Really Learn Lexical Inference Relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, CO, USA. Association for Computational Linguistics.

Lin, D. and Pantel, P. (2001). Induction of Semantic Classes from Natural Language Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 317–322, San Francisco, CA, USA. ACM.

Meyer, C. M., Mieskes, M., Stab, C., and Gurevych, I. (2014). DKPro Agreement: An Open-Source Java Library for Measuring Inter-Rater Agreement. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 105–109, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Mirkin, S., Dagan, I., and Geffet, M. (2006). Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 579–586, Sydney, Australia. Association for Computational Linguistics.

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

Necsulescu, S., Mendes, S., Jurgens, D., Bel, N., and Navigli, R. (2015). Reading Between the Lines: Overcoming Data Sparsity for Accurate Classification of Lexical Relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 182–192, Denver, CO, USA. Association for Computational Linguistics.

Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (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.

Panchenko, A., Faralli, S., Ruppert, E., Remus, S., Naets, H., Fairon, C., Ponzetto, S. P., and Biemann, C. (2016). TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327, San Diego, California, USA. Association for Computational Linguistics.

Panchenko, A., Marten, F., Ruppert, E., Faralli, S., Ustalov, D., Ponzetto, S. P., and Biemann, C. (2017). Unsupervised, knowledge-free, and interpretable word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 91–96, Copenhagen, Denmark, September. Association for Computational Linguistics.

Pantel, P. and Lin, D. (2002). Discovering Word Senses from Text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 613–619, Edmonton, AB, Canada. ACM.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia. Association for Computational Linguistics.

Pantel, P. and Ravichandran, D. (2004). Automatically Labeling Semantic Classes. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'2004)*, pages 321–328, Boston, MA, USA. Association for Computational Linguistics.

Randolph, J. J. (2005). Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Online submission*.

Richter, M., Quasthoff, U., Hallsteinsdóttir, E., and Biemann, C. (2006). Exploiting the Leipzig Corpora Collection. In *Proceedings of the Fifth Slovenian and First International Language Technologies Conference (IS-LTC)*, Ljubljana, Slovenia. European Language Resources Association (ELRA).

Roller, S., Erk, K., and Boleda, G. (2014). Inclusive yet Selective: Supervised Distributional Hypernymy Detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., and Ponzetto, S. P. (2016). A Large DataBase of Hypernymy Relations Extracted from the Web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, LREC 2016, pages 360–367, Portorož, Slovenia. European Language Resources Association (ELRA).

Shi, L. and Mihalcea, R. (2005). Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing 2005, pages 100–111, Mexico City, Mexico. Springer Berlin Heidelberg.

Shwartz, V., Goldberg, Y., and Dagan, I. (2016). Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany, August. Association for Computational Linguistics.

Snow, R., Jurafsky, D., and Ng, A. Y. (2004). Learning Syntactic Patterns for Automatic Hypernym Discovery.

In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 1297–1304, Vancouver, BC, Canada. MIT Press.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *arXiv preprint cs/0609058*.

Steyvers, M. and Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive science*, 29(1):41–78.

Tjong Kim Sang, E. (2007). Extracting Hypernym Pairs from the Web. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 165–168, Prague, Czech Republic. Association for Computational Linguistics.

Ustalov, D., Arefyev, N., Biemann, C., and Panchenko, A. (2017a). Negative sampling improves hypernymy extraction based on projection learning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 543–550, Valencia, Spain. Association for Computational Linguistics.

Ustalov, D., Panchenko, A., and Biemann, C. (2017b). Watset: Automatic induction of synsets from a graph of synonyms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1579–1590, Vancouver, Canada. Association for Computational Linguistics.

Velardi, P., Faralli, S., and Navigli, R. (2013). OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction. *Computational Linguistics*, 39(3):665–707.

Vylomova, E., Rimell, L., Cohn, T., and Baldwin, T. (2016). Take and Took, Gaggle and Goose, Book and Read: Evaluating the Utility of Vector Differences for Lexical Relation Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany. Association for Computational Linguistics.

Wandmacher, T. (2005). How semantic is Latent Semantic Analysis? In *Proceedings of RÉCITAL 2005*, pages 525–534, Dourdan, France.

Weeds, J., Clarke, D., Reffin, J., Weir, D. J., and Keller, B. (2014). Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Zhou, G., Liu, Y., Liu, F., Zeng, D., and Zhao, J. (2013). Improving question retrieval in community question answering using world knowledge. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2239–2245, Beijing, China. AAAI Press.