Unsupervised Sense-Aware Hypernymy Extraction

Dmitry Ustalov[†], **Alexander Panchenko**[‡],
Chris Biemann[‡], and **Simone Paolo Ponzetto**[†]

[†]University of Mannheim, Germany
{dmitry,simone}@informatik.uni-mannheim.de
[‡]University of Hamburg, Germany
{panchenko,biemann}@informatik.uni-hamburg.de

Abstract

In this paper, we show how unsupervised sense representations can be used to improve hypernymy extraction. We present a method for extracting disambiguated hypernymy relationships that propagate hypernyms to sets of synonyms (synsets), constructs embeddings for these sets, and establishes sense-aware relationships between matching synsets. Evaluation on two gold standard datasets for English and Russian shows that the method successfully recognizes hypernymy relationships that cannot be found with standard Hearst patterns and Wiktionary datasets for the respective languages.

1 Introduction

Hypernymy relationships are of central importance in natural language processing. They can be used to automatically construct taxonomies (Bordea et al., 2016; Faralli et al., 2017; Faralli et al., 2018), expand search engine queries (Gong et al., 2005), improve semantic role labeling (Shi and Mihalcea, 2005), perform generalizations of entities mentioned in questions (Zhou et al., 2013), and so forth. One of the important use cases of hypernyms is lexical expansion as in the following sentence: "This bar serves fresh *jabuticaba* juice". Representation of the rare word "jabuticaba" can be noisy, yet it can be substituted by its hypernym "fruit", which is frequent and has a related meaning. Note that, in this case, subword information provided by character-based distributional models, such as *fastText* (Bojanowski et al., 2017), does not help to derive the meaning of the rare word.

Currently available hypernymy extraction methods perform extraction of hypernymy relationships from text between two ambiguous words, e.g., apple $\succ$ fruit. However, by definition in Cruse (1986), hypernymy is a binary relationship between senses, e.g., $apple^2 \succ fruit^1$, where $apple^2$ is the "food" sense of the word "apple". In turn, the word "apple" can be represented by multiple lexical units, e.g., "apple" or "pomiculture". This sense is distinct from the "company" sense of the word "apple", which can be denoted as $apple^3$. Thus, more generally, hypernymy is a relation defined on two sets of disambiguated words; this modeling principle was also implemented in WordNet (Fellbaum, 1998), where hypernymy relations link not words directly, but instead synsets. This essential property of hypernymy is however not used or modeled in the majority of current hypernymy extraction approaches. In this paper, we present an approach that addresses this shortcoming.

The contribution of our work is a novel approach that, given a database of noisy ambiguous hypernyms, (1) removes incorrect hypernyms and adds missing ones, and (2) disambiguates related words. Our unsupervised method relies on synsets induced automatically from synonymy dictionaries. In contrast to prior approaches, such as the one by Pennacchiotti and Pantel (2006), our method not only disambiguates the hypernyms but also extracts new relationships, substantially improving F-score over the original extraction in the input collection of hypernyms. We are the first to use sense representations to improve hypernymy extraction, as opposed to prior art.

2 Related Work

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. All of

Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)
Vienna, Austria – September 19-21, 2018

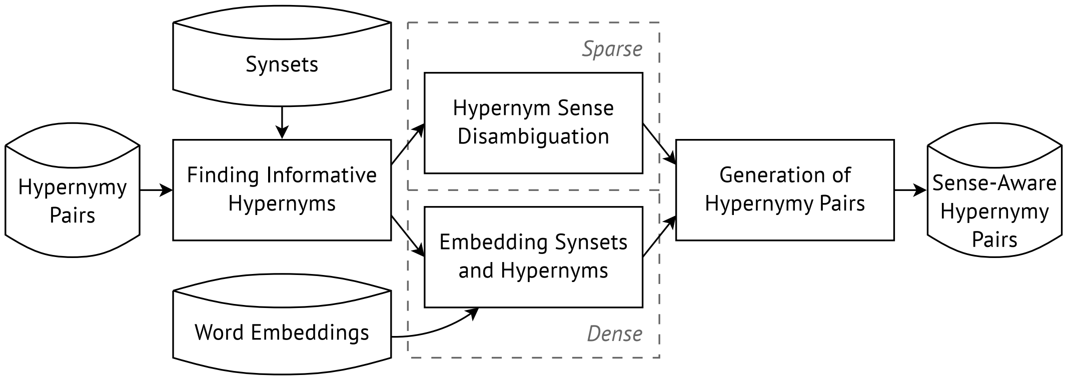

Figure 1: Outline of the proposed method for sense-aware hypernymy extraction using synsets.

these approaches use a small set of training hypernymy pairs to bootstrap the pattern discovery process. Tjong Kim Sang (2007) used Web snippets as a corpus for a similar approach. More recent approaches exploring the use of distributional word representations for extraction of 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 relationship between two words, e.g., on the basis of the difference of such vectors or their concatenation.

Recent approaches to hypernym extraction went into three directions: (1) unsupervised methods based on such huge corpora as CommonCrawl[1] to ensure extraction coverage using Hearst (1992) patterns (Seitner et al., 2016); (2) learning patterns in a supervised way based on a combination of syntactic patterns and distributional features in the HypeNet model (Shwartz et al., 2016); (3) transforming (Ustalov et al., 2017a) or specializing (Glavaš and Ponzetto, 2017) word embedding models to ensure the property of asymmetry. We tested our method based on a large-scale database of hypernyms extracted in an unsupervised way using Hearst patterns. While methods, such as those by Mirkin et al. (2006), Shwartz et al. (2016), Ustalov et al. (2017a) and Glavaš and Ponzetto (2017) use distributional features for extraction of hypernyms, they do not take into account word sense representations: this is despite hypernymy being a semantic relation holding between senses.

The only sense-aware approach we are aware of is presented by Pennacchiotti and Pantel (2006). Given a set of extracted binary semantic relationships, this approach disambiguates them with respect to the WordNet sense inventory (Fellbaum, 1998). In contrast to our work, the authors do not use the synsets to improve the coverage of the extracted relationships.

Note that we propose an approach for post-processing of hypernyms based on a model of distributional semantics. Therefore, it can be applied to any collection of hypernyms, e.g., extracted using Hearst patterns, HypeNet, etc. Since our approach outputs dense vector representations for synsets, it could be useful for addressing such tasks as knowledge base completion (Bordes et al., 2011).

3 Using Synsets for Sense-Aware Hypernymy Extraction

We use the sets of synonyms (synsets) expressed in such electronic lexical databases as WordNet (Fellbaum, 1998) to disambiguate the words in extracted hyponym-hypernym pairs. We also use synsets to propagate the hypernymy relationships to the relevant words not covered during hypernymy extraction.

Our unsupervised method, shown in Figure 1, relies on the assumption that the words in a synset have similar hypernyms (Section 3.1). We exploit this assumption to gather all the possible hypernyms for a synset and rank them according to their importance (Section 3.2). Then, we disambiguate the hypernyms, i.e., for each hypernym, we find the sense which synset maximizes the similarity to the set of gathered hypernyms (Section 3.3).

Additionally, we use distributional word representations to transform the sparse synset representations into dense synset representations. We obtain such representations by aggregating the word embeddings corresponding to the elements of synsets and sets of hypernyms (Section 3.4). Finally, we generate the sense-aware hyponym-hypernym pairs

[1] `https://commoncrawl.org`

Algorithm 1 Unsupervised Sense-Aware Hypernymy Extraction.

Input: a vocabulary V, a set of word senses $\mathcal{V}$, a set of synsets $\mathcal{S}$, a set of *is-a* pairs $R \subset V^2$.
a number of top-scored hypernyms $n \in \mathbb{N}$,
a number of nearest neighbors $k \in \mathbb{N}$,
a maximum matched synset size $m \in \mathbb{N}$.

Output: a set of sense-aware *is-a* pairs $\mathcal{R} \subset \mathcal{V}^2$.

for all $S \in \mathcal{S}$ **do**
  $\text{label}(S) \leftarrow \{h \in V : (w,h) \in R, w \in \text{words}(S)\}$
for all $S \in \mathcal{S}$ **do**
  for all $h \in \text{label}(S)$ **do**
    $\text{tf–idf}(h,S,\mathcal{S}) \leftarrow \text{tf}(h,S) \times \text{idf}(h,\mathcal{S})$
for all $S \in \mathcal{S}$ **do** // Hypernym Sense Disambiguation
  $\widehat{\text{label}}(S) \leftarrow \emptyset$
  for all $h \in \text{label}(S)$ **do** // Take only top-n elements of label(S)
    $\hat{S} \leftarrow \arg\max_{S' \in \mathcal{S}:\text{senses}(h) \cap S' \neq \emptyset} \text{sim}(\text{label}(S), \text{words}(S'))$
    $\hat{h} \leftarrow \text{senses}(h) \cap \hat{S}$
    $\widehat{\text{label}}(S) \leftarrow \widehat{\text{label}}(S) \cup \{\hat{h}\}$
for all $S \in \mathcal{S}$ **do** // Embedding Synsets and Hypernyms
  $\vec{S} \leftarrow \frac{\sum_{w \in \text{words}(S)} \vec{w}}{|S|}$
  $\overrightarrow{\text{label}}(S) \leftarrow \frac{\sum_{h \in \text{label}(S)} \text{tf–idf}(h,S,\mathcal{S}) \cdot \vec{h}}{\sum_{h \in \text{label}(S)} \text{tf–idf}(h,S,\mathcal{S})}$
  $\hat{S} \leftarrow \arg\max_{S' \in \text{NN}_k(\overrightarrow{\text{label}}(S)) \cap \mathcal{S} \setminus \{S\}} \text{sim}(\overrightarrow{\text{label}}(S), \vec{S'})$
  if $|\hat{S}| \leq m$ **then**
    $\widehat{\text{label}}(S) \leftarrow \widehat{\text{label}}(S) \cup \hat{S}$
return $\bigcup_{S \in \mathcal{S}} S \times \widehat{\text{label}}(S)$

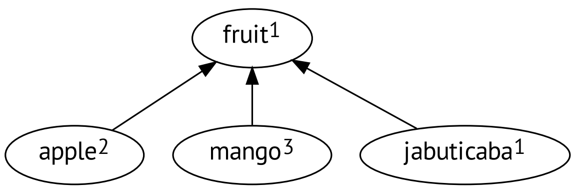

Figure 2: Disambiguated hypernymy relationships: each hypernym has a sense identifier from the predefined sense inventory.

by computing cross products (Section 3.5).

Let V be a vocabulary of ambiguous words, i.e., a set of all lexical units (words) in a language. Let $\mathcal{V}$ be a set of all the senses for the words in V. For instance, $apple^2 \in \mathcal{V}$ is a sense of $apple \in V$. For simplicity, we denote $\text{senses}(w) \subseteq \mathcal{V}$ as the set of sense identifiers for each word $w \in V$. Then, we define a synset $S \in \mathcal{S}$ as a subset of $\mathcal{V}$.

Given a vocabulary V, we denote the input set of *is-a* relationships as $R \subset V^2$. This set is provided in the form of tuples $(w,h) \in R$. Given the nature of our data, we treat the terms *hyponym* $w \in V$ and *hypernym* $h \in V$ in the lexicographical meaning. These lexical units have no sense labels attached, e.g., $R = \{(cherry, color), (cherry, fruit)\}$. Thus, given a set of synsets $\mathcal{S}$ and a relation $R \subset V^2$, our goal is to construct an asymmetrical relation $\mathcal{R} \subset \mathcal{V}^2$ that represents meaningful hypernymy relationships between word *senses*.

The complete pseudocode for the proposed approach is presented in Algorithm 1; the output of the algorithm is the sense-aware hypernymy relation $\mathcal{R}$ (cf. Figure 2). The following sections describe various specific aspects of the approach.

3.1 Obtaining Synsets

A synset is a linguistic structure which is composed of a set of mutual synonyms, all representing the same word sense. For instance, WordNet described two following senses of the word "mango", which correspond to a tree and a fruit respectively, as illustrated in Figure 3. Note that, depending on the word sense, the word "mango" can have a different hypernym, which is also a synset in turn.

In our experiments, presented in this paper, we rely on synsets from the manually constructed lexical resources, such as WordNet (Fellbaum, 1998),

Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)
Vienna, Austria – September 19-21, 2018

- S: (n) **mango**, mango tree, Mangifera indica (large evergreen tropical tree cultivated for its large oval fruit)
    - part meronym
    - member holonym
    - direct hypernym / inherited hypernym / sister term
        - S: (n) fruit tree (tree bearing edible fruit)
- S: (n) **mango** (large oval tropical fruit having smooth skin, juicy aromatic pulp, and a large hairy seed)
    - ***direct hypernym*** / *inherited hypernym* / *sister term*
        - S: (n) edible fruit (edible reproductive body of a seed plant especially one having sweet flesh)
    - *part holonym*

Figure 3: Synsets of the word "mango" from the Princeton WordNet and their respective hypernyms.

and on synsets constructed automatically from synonymy dictionaries, using the WATSET algorithm (Ustalov et al., 2017b).

While synonymy dictionaries can be extracted from Wiktionary and similar resources for almost any language, coverage of such dictionaries, for some languages can be still scarce. For these cases, instead of synsets, our approach can be used with distributionally induced word senses extracted from unlabelled text corpora. We explored this route in (Panchenko et al., 2018).

3.2 Finding Informative Synset Hypernyms

We start with finding informative hypernyms for every synset. In real-world datasets, the input relation R can contain noise in the form of mistakenly retrieved co-occurrences and various human errors. In order to get rid of these mistakes, we map every synset $S \in \mathcal{S}$ to a *bag of words* $\operatorname{label}(S) \subset V$ without sense identifiers. This synset label holds a bag of hypernyms in R matching the words in S as hyponyms in lines 1–2:

$$\operatorname{label}(S) = \{h \in V : (w,h) \in R, w \in \operatorname{words}(S)\}. \quad (1)$$

In case the relation R is provided with the counts of pair occurrences in a corpus, we add each occurrence into $\operatorname{label}(S)$. Furthermore, since $\operatorname{label}(S)$ is a bag allowing multiple occurrences of the same hypernyms for different words included to the synset, we model the variable importance of words in labels using the tf–idf weighing scheme (Salton and Buckley, 1988) in lines 3–5:

$$\text{tf–idf}(h,S,\mathcal{S}) = \operatorname{tf}(h,S) \times \operatorname{idf}(h,\mathcal{S}), \quad (2)$$

$$\operatorname{tf}(h,S) = \frac{|h' \in \operatorname{label}(S) : h = h'|}{|\operatorname{label}(S)|}, \quad (3)$$

$$\operatorname{idf}(h,\mathcal{S}) = \log \frac{|\mathcal{S}|}{|S' \in \mathcal{S} : h \in \operatorname{label}(S')|}. \quad (4)$$

In order to ensure that the most important hypernyms are the terms that often were identified as hypernyms for the respective synset, we limit the maximal size of $\operatorname{label}(S)$ to a parameter $n \in \mathbb{N}$. As the result of this step, each synset is provided with a set of top-n hypernyms the importance of which is measured using tf–idf.

3.3 Hypernym Sense Disambiguation

The words in the synset labels are not yet provided with sense labels, so in this step, we run a word sense disambiguation procedure that is similar to the one by Faralli et al. (2016). In particular, given a synset $S \in \mathcal{S}$ and its $\operatorname{label}(S) \subseteq V$, for each hypernym $h \in \operatorname{label}(S)$ we aim at finding the synset $S' \in \mathcal{S}$ such that it is similar to the whole $\operatorname{label}(S)$ containing this hypernym while it is not equal to S.

We perform the hypernym sense disambiguation as follows. Every synset and every label are represented as sparse vectors in a vector space model that enables computing distances between the vectors (Salton et al., 1975). Given a synset $S \in \mathcal{S}$ and its label, for each hypernym $h \in \operatorname{label}(S)$ we iterate over all the synsets that include h as a word. We maximize the cosine similarity measure between $\operatorname{label}(S)$ and the candidate synset $S' \in \mathcal{S}$ to find the synset $\hat{S}$ the meaning of which is the most similar to $\operatorname{label}(S)$. The following procedure is used (line 9):

$$\hat{S} = \operatorname*{arg\,max}_{S' \in \mathcal{S} : \operatorname{senses}(h) \cap S' \neq \emptyset} \operatorname{sim}(\operatorname{label}(S), \operatorname{words}(S')). \quad (5)$$

Having obtained the synset $\hat{S}$ that is closest to $\operatorname{label}(S)$, we treat $\hat{h} = \operatorname{senses}(h) \cap \hat{S}$ as the desired disambiguated sense of the hypernym $h \in \operatorname{label}(S)$. This procedure is executed for every word in the label to produce a disambiguated label (lines 10–11):

$$\widehat{\operatorname{label}}(S) = \{\hat{h} \in \mathcal{V} : h \in \operatorname{label}(S)\} \quad (6)$$

The result of the label construction step is the set of disambiguated hypernyms linked to each synset. For example, consider the hypernymy label {*fruit*,*food*,*cherry*} and two following synsets: $\{cherry^1, red\,fruit^1, fruit^1\}$ and $\{cherry^2, cerise^1, cherry\ red^1\}$. The disambiguation procedure will choose the first sense of the word "fruit" in the hypernymy label because the latter synset is more similar to the given label.

3.4 Embedding Synsets and Hypernyms

In order to overcome data sparsity by retrieving more relevant senses, we use such distributional word representations as Skip-gram (Mikolov et al., 2013). We embed synsets and their labels in a low-dimensional vector space to perform matching. This matching makes it possible to produce more sense-aware hypernymy pairs as it captures the hierarchical relationships between synsets through their labels. Given a word $w \in V$, we denote as $\vec{w} \in \mathbb{R}^d$ a d-dimensional vector representation of this word.

Given the empirical evidence of the fact that a simple averaging of word embeddings yields a reasonable vector representation (Socher et al., 2013), we follow the SenseGram approach by Pelevina et al. (2016) to compute synset embeddings. We perform *unweighted* pooling as the words constituting synsets are equally important (line 13):

$$\vec{S} = \frac{\sum_{w \in \mathrm{words}(S)} \vec{w}}{|S|}. \tag{7}$$

In contrast to the approach we use to embed synsets, we perform *weighted* pooling of the word embeddings to compute the label embeddings. Like the weights, we use tf–idf scores produced at the synset labeling stage (Section 3.2). Thus, each $\mathrm{label}(S)$ is mapped to the following low-dimensional vector (line 14):

$$\overrightarrow{\mathrm{label}}(S) = \frac{\sum_{h \in \mathrm{label}(S)} \text{tf–idf}(h, S, \mathcal{S}) \cdot \vec{h}}{\sum_{h \in \mathrm{label}(S)} \text{tf–idf}(h, S, \mathcal{S})}. \tag{8}$$

Now, we use a *top-down* procedure for establishing relationships between the synsets as follows. We represent all the synsets $\mathcal{S}$ and all their labels in *the same* vector space. Then, for each synset label, we search for the $k \in \mathbb{N}$ nearest neighbors of the label vector. In case we find a synset among the top neighbors, we treat it as the set of hypernyms of the given synset. Specifically, given a synset $S \in \mathcal{S}$ and its $\overrightarrow{\mathrm{label}}(S) \in \mathbb{R}^d$, we extract a set of nearest neighbors $\mathrm{NN}_k(\overrightarrow{\mathrm{label}}(S))$. Each element of the result set can be either a synset or a label. We do not take into account the neighbors that are labels. We also exclude the input synset from the result set. Thus, for the synset $\hat{S}$ we use a disambiguation procedure shown in line 15:

$$\hat{S} = \operatorname*{arg\,max}_{S' \in \mathrm{NN}_k(\overrightarrow{\mathrm{label}}(S)) \cap \mathcal{S} \setminus \{S\}} \mathrm{sim}(\overrightarrow{\mathrm{label}}(S), \vec{S'}). \tag{9}$$

Additionally, we require that no candidate synset includes more than $m \in \mathbb{N}$ words as it can hardly represent a reasonable set of synonyms. Finally, to each $S \in \mathcal{S}$ we assign $\widehat{\mathrm{label}}(S) = \widehat{\mathrm{label}}(S) \cup \hat{S}$ in lines 16–17. In case no synsets are found, we skip S. During prototyping, we tried the bottom-up procedure of searching a label given a synset. Our experiments showed that such a procedure is inefficient and fails to provide a reasonable matching.

3.5 Generation of Hypernymy Pairs

We generate an output set of sense-aware hyponym-hypernym pairs $\mathcal{R} \subset \mathcal{V}^2$ by computing a cross product between the set of synsets and the set the labels corresponding to them (line 18):

$$\mathcal{R} = \bigcup_{S \in \mathcal{S}} S \times \widehat{\mathrm{label}}(S). \tag{10}$$

As the result, the example in Figure 2 will be transformed into the following relation $\mathcal{R}$:

Hyponym Sense	Hypernym Sense
apple^2	*fruit*^{1}
mango^3	*fruit*^{1}
jabuticaba^1	*fruit*^{1}

4 Evaluation

We conduct two experiments based on well-known gold standards to address the following research questions:

RQ1 How well does the proposed approach generalize the hypernyms given *the synsets of the gold standard*?

RQ2 How well does the proposed approach generalize the hypernyms given *the synsets not belonging to the gold standard*?

We run our experiments on two different languages, namely English, for which a large amount of lexical semantic resources are available, and Russian, which is an under-resourced natural language.

Table 1: Hypernyms used to construct labels of the input synsets, the frequency threshold for Hearst Patterns is denoted as $f \in \mathbb{N}$.

Language	Name	# pairs
English	Wiktionary	62 866
	Hearst Patterns ($f \geq 100$)	39 650
	ALL (Wiktionary + Hearst Patterns)	102 516
Russian	Wiktionary	185 257
	Hearst Patterns ($f \geq 30$)	10 458
	Small Academic Dictionary	38 661
	ALL (Wiktionary + Small Academic Dictionary + Hearst Patterns)	234 376

We report the performance of two configurations of our approach. The first configuration, *Sparse*, excludes the embedding approach described in Section 3.4 (lines 13–17). The second configuration, *Full*, is a complete setup of our approach, which includes the relation extracted with the *Sparse* configuration and further extends them with relations extracted using synset-hypernym embedding matching mechanism.

4.1 Experimental Setup

Given a gold standard taxonomy, composed of hypernymy relations, one can evaluate the quality of the automatically extracted hypernyms by comparing them to this resource. A common evaluation measure for assessing taxonomies is the cumulative Fowlkes–Mallows index proposed by Velardi et al. (2013). However, this measure cannot be applied for relatively large graphs like ours due to running a depth-first search (DFS) algorithm to split the input directed graph into levels. Since our graphs have hundreds of thousands of nodes (cf. Table 1), this approach is not tractable in reasonable time unlike in the evaluation by Bordea et al. (2016) that was applied to much smaller graphs. To make our evaluation possible, we perform directed path existence checks in the graphs instead of the DFS algorithm execution. In particular, we rely on precision, recall, F-score w.r.t. a sense-aware gold standard set of hypernyms. For that, sense labels are removed from the compared methods and then an *is-a* pair $(w,h) \in R$ is considered as predicted correctly *if and only if there is a path from some sense of w to some sense of h in the gold standard dataset*. Let $G = (V_G, E_G)$ be the gold standard taxonomy and $H = (V, E)$ be the taxonomy to evaluate against G. Let $u \overset{G}{\to} v$ be the directed path from the node u to the node v in G. Then, we define the numbers of positive and negative answers as follows:

Table 2: Skip-gram-based word embeddings used to construct synset embeddings.

Language	Dataset	Genre	Dim.	# tokens
English	Google News	news	300	100×10^9
Russian	RDT	books	500	13×10^9

$$\mathrm{TP} = |(u,v) \in E \ : \exists u \overset{G}{\to} v|, \tag{11}$$

$$\mathrm{FP} = |(u,v) \in E \ : \nexists u \overset{G}{\to} v|, \tag{12}$$

$$\mathrm{FN} = |(u,v) \in E_G \colon \nexists u \overset{H}{\to} v|, \tag{13}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. As the result, we use the standard definitions of precision as $\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$, recall as $\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$, and F-score as $\mathrm{F}_1 = \frac{2 \cdot \mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr}+\mathrm{Re}}$.

Note that the presented approach could overestimate the number of true positives when the nodes are located far from each other in the gold standard. Only the words appearing both in the gold standard and in the comparable datasets are considered. The remaining words are excluded from the evaluation.

4.2 Datasets

The hypernymy datasets for both languages have been extracted from Wiktionary using the JWKTL tool by Zesch et al. (2008); the Wiktionary dump was obtained on June 1, 2018. As the non-gold datasets of synsets, we use the automatically discovered synsets published by Ustalov et al. (2017b) for both English and Russian.[2]

For *English*, we combine two data sources: *Wiktionary* and a hypernymy pair dataset obtained

[2] `https://github.com/dustalov/watset/releases/tag/v1.0`

using *Hearst Patterns* from a large text corpus. The corpus has 9 billion tokens compiled from the Wikipedia[3], Gigaword (Graff and Cieri, 2003), and ukWaC (Ferraresi et al., 2008) corpora. The union of hypernyms from Wiktionary and Hearst patterns is denoted as ALL. As word embeddings for English, we use the Google News vectors.[4] Finally, WordNet (Fellbaum, 1998) was used as the gold standard dataset in our experiments as a commonly used source of ground truth hypernyms.

For *Russian*, we use a composition of three different hypernymy pair datasets summarized in Table 1: a dataset extracted from the `lib.rus.ec` electronic library using the Hearst (1992) patterns implemented for the Russian language in the PatternSim[5] toolkit (Panchenko et al., 2012), a dataset extracted from the Russian *Wiktionary*, and a dataset extracted from the sense definitions in the *Small Academic Dictionary* (SAD) of the Russian language (Kiselev et al., 2015). We also consider the ALL dataset uniting *Patterns*, *Wiktionary* and *Small Academic Dictionary*. As word embeddings, we use the Russian Distributional Thesaurus (RDT) vectors.[6] Finally, as the gold standard, we use the RuWordNet[7] lexical database for Russian (Loukachevitch et al., 2016).

4.3 Meta-Parameters of the Methods

Parameter tuning during prototyping showed that the optimal parameters for English were $n = 3$, $k = 1$ and $m = 15$ for WordNet, and $n = 3$, $k = 1$ and $m = 20$ for WATSET; for Russian the optimal values were $n = 3$, $k = 1$ and $m = 20$ for all the cases. Table 2 briefly describes the word embedding datasets.

5 Results and Discussion

Tables 3 and 4 show the results for the first experiment on hypernymy extraction using for both languages. According to the experimental results for both languages on the *gold standard synsets*, the *Full* model outperforms the others in terms of recall and F-score. The improvements are due to gains in recall with respect to the input hypernyms (*No Synsets*). This confirms that the proposed approach

Table 3: Performance of our methods on the WordNet gold standard using the synsets from WordNet (PWN) and automatically induced synsets (WATSET) for English; the best overall results are boldfaced.

	Method	# pairs	Pr	Re	F_1
PWN	*Full*(ALL)	75894	53.23	**39.95**	**45.27**
PWN	*Sparse*(ALL)	61056	56.78	36.72	44.60
WATSET	*Full*(ALL)	72686	57.60	18.93	28.49
WATSET	*Sparse*(ALL)	40303	62.42	16.85	26.53
No Synsets	ALL	98096	64.84	18.72	29.05
No Synsets	Hearst Patterns	38530	**67.09**	16.57	26.58
No Synsets	Wiktionary	59674	46.78	1.36	2.64

improves the quality of the input hypernymy pairs by correctly propagating the hypernymy relationships to previously non-covered words with the same meaning.

According to the experiments on the *automatically induced synsets* by the WATSET method from Ustalov et al. (2017b), the *Full* model also yields the best results, the quality of the synset embeddings greatly depends on the quality of the corresponding synsets. While these synsets did not improve the quality of the hypernymy extraction for English, they show large gains for Russian.

Error analysis shows the improvements for Russian can be explained by higher quality input synsets for this language: some English synsets are implausible according to our human judgment. For both languages, our method improves both precision and recall compared to the union of the input hypernyms, ALL. Finally note that while the absolute numbers of precision and recall are somewhat low, especially for the Russian language, these performance scores are low even for resources constructed completely manually, e.g., Wiktionary and the Small Academic Dictionary in Table 4. This is the result of a vocabulary mismatch between the gold standards and the input hypernymy datasets. Note that the numbers of pairs reported in Tables 3 and 4 differ from the numbers presented in Table 1 also due to a vocabulary mismatch.

6 Conclusion

In this study, we presented an unsupervised method for disambiguation and denoising of an input database of noisy ambiguous hypernyms using automatically induced synsets. Our experiments show a substantial performance boost on both gold stan-

[3]http://panchenko.me/data/joint/corpora/en59g/wikipedia.txt.gz
[4]https://code.google.com/archive/p/word2vec/
[5]https://github.com/cental/patternsim
[6]https://russe.nlpub.org/downloads/
[7]http://ruwordnet.ru/en/

Table 4: Performance of our methods on the RuWordNet gold standard using the synsets from RuWordNet (RWN) and automatically induced synsets (WATSET) for Russian; the best overall results are boldfaced.

	Method	# pairs	Pr	Re	F_1
RWN	*Full*(ALL)	297 387	37.65	**41.88**	**39.65**
	Sparse(ALL)	145 114	31.53	22.02	25.93
WATSET	*Full*(ALL)	281 006	25.75	17.27	20.67
	Sparse(ALL)	166 937	25.58	13.83	17.95
No Synsets	ALL	212 766	23.48	9.81	13.84
	SAD	36 800	24.41	5.44	8.90
	Wiktionary	172 999	**42.04**	3.78	6.94
	Hearst Patterns	10 458	39.49	0.62	1.22

dard datasets for English and Russian on a hypernymy extraction task. Especially supported by our results on Russian, we conclude that our approach, provided even with a set of automatically induced synsets, improves hypernymy extraction without explicit human input. The implementation[8] of the proposed approach and the induced resources[9] are available online. Possible directions for future studies include using a different approach for synset embeddings (Rothe and Schütze, 2015) and hypernym embeddings (Nickel and Kiela, 2017).

Acknowledgments

We acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) foundation under the "JOIN-T" and "ACQuA" projects, and the Deutscher Akademischer Austauschdienst (DAAD). Finally, we are grateful to three anonymous reviewers for providing valuable comments.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. SemEval-2016 Task 13: Taxonomy Extraction Evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091, San Diego, CA, USA. Association for Computational Linguistics.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-11)*, pages 301–306, San Francisco, CA, USA. Association for the Advancement of Artificial Intelligence.

Alan Cruse. 1986. *Lexical Semantics*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Stefano Faralli, Alexander Panchenko, Chris Biemann, and Simone Paolo Ponzetto. 2016. Linked Disambiguated Distributional Semantic Networks. In *The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Proceedings, Part II*, pages 56–64, Kobe, Japan. Springer International Publishing.

Stefano Faralli, Alexander Panchenko, Chris Biemann, and Simone Paolo Ponzetto. 2017. The ContrastMedium Algorithm: Taxonomy Induction From Noisy Knowledge Graphs With Just A Few Links. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, EACL 2017, pages 590–600, Valencia, Spain. Association for Computational Linguistics.

Stefano Faralli, Irene Finocchi, Simone Paolo Ponzetto, and Paola Velardi. 2018. Efficient Pruning of Large Knowledge Graphs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI-18, pages 4055–4063, Stockholm, Sweden. International Joint Conferences on Artificial Intelligence Organization.

Christiane Fellbaum. 1998. *WordNet: An Electronic Database*. MIT Press.

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, Marrakech, Morocco.

Goran Glavaš and Simone Paolo Ponzetto. 2017. Dual Tensor Model for Detecting Asymmetric Lexico-Semantic Relations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2017, pages 1757–1767, Copenhagen, Denmark. Association for Computational Linguistics.

Zhiguo Gong, Chan Wa Cheang, 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. Springer Berlin Heidelberg, Copenhagen, Denmark.

David Graff and Christopher Cieri. 2003. English Gigaword. https://catalog.ldc.upenn.edu/ldc2003t05.

[8] https://github.com/dustalov/watlink, https://github.com/tudarmstadt-lt/sensegram/blob/master/hypernyms.ipynb

[9] http://ltdata1.informatik.uni-hamburg.de/joint/hyperwatset/konvens

Marti A. Hearst. 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.

Yuri Kiselev, Sergey V. Porshnev, and Mikhail Mukhin. 2015. Method of Extracting Hyponym-Hypernym Relationships for Nouns from Definitions of Explanatory Dictionaries. *Programmnaya Ingeneriya*, 10:38–48. In Russian.

Natalia V. Loukachevitch, German Lashevich, Anastasia A. Gerasimova, Vladimir V. Ivanov, and Boris V. Dobrov. 2016. Creating Russian WordNet by Conversion. In *Computational Linguistics and Intellectual Technologies: papers from the Annual conference "Dialogue"*, pages 405–415, Moscow, Russia. RSUH.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., Harrahs and Harveys, NV, USA.

Shachar Mirkin, Ido Dagan, and Maayan Geffet. 2006. Integrating Pattern-based and Distributional Similarity Methods for Lexical Entailment Acquisition. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 579–586, Sydney, NSW, Australia. Association for Computational Linguistics.

Silvia Necsulescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 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.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems 30*, NIPS 2017, pages 6338–6347. Curran Associates, Inc., Long Beach, CA, USA.

Alexander Panchenko, Olga Morozova, and Hubert Naets. 2012. A Semantic Similarity Measure Based on Lexico-Syntactic Patterns. In *Proceedings of the 11th Conference on Natural Language Processing (KONVENS 2012)*, pages 174–178, Vienna, Austria. ÖGAI.

Alexander Panchenko, Dmitry Ustalov, Stefano Faralli, Simone Paolo Ponzetto, and Chris Biemann. 2018. Improving Hypernymy Extraction with Distributional Semantic Classes. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC 2018, pages 1541–1551, Miyazaki, Japan. European Language Resources Association (ELRA).

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

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

Marco Pennacchiotti and Patrick Pantel. 2006. Ontologizing Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 793–800, Sydney, NSW, Australia. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 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.

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

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Gerard Salton, Andrew Wong, and Chungshu S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620.

Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 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).

Lei Shi and Rada Mihalcea. 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.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 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. Association for Computational Linguistics.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2013, pages 1631–1642, Seattle, WA, USA. Association for Computational Linguistics.

Erik Tjong Kim Sang. 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.

Dmitry Ustalov, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. 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*, EACL 2017, pages 543–550, Valencia, Spain. Association for Computational Linguistics.

Dmitry Ustalov, Alexander Panchenko, and Chris Biemann. 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, BC, Canada. Association for Computational Linguistics.

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

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 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.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 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.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, LREC 2008, pages 1646–1652, Marrakech, Morocco. European Language Resources Association (ELRA).

Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. 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 / International Joint Conferences on Artificial Intelligence.