Proceedings of the 12th Web as Corpus Workshop, pages 42–49
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Hypernym-LIBre: A free Web-based Corpus for Hypernym Detection

Shaurya Rawat, Mariano Rico, Oscar Corcho
Ontology Engineering Group
Universidad Politécnica de Madrid, Madrid, Spain
{srawat@delicias.dia.fi., mariano.rico@, ocorcho@fi.} upm.es

Abstract

We describe a web-based corpus for hypernym detection which consists of 32 GB of high quality English paragraphs along with their part-of-speech tagged and dependency parsed versions. One of the main advantages of this corpus is that it is available under an open license while providing similar results for training and testing on state-of-the-art methods and techniques for detecting hypernyms, which makes it a good alternative to currently used corpora which are not available freely. The corpus has been created by cleaning and pre-processing the existing UMBC web-corpus and English Wikipedia.We detail existing methods for hypernym detection and analyze the state-of-the-art techniques using our corpus as a text source. We evaluate the corpus using 5 datasets and 4 models and compare them.

Keywords: hypernym detection, NLP, web-based corpus, Hearst patterns

1. Introduction

Hyponyms are terms whose semantic field lies within that of another term, which is called its Hypernym. They capture the 'is-a' or 'type-of' relationship between terms. It is also sometimes referred to as the *umbrella term* or the *blanket term*. For example: "Spain is a country". In this case, 'Spain' is an instance of the type 'country' and therefore 'country' is its hypernym. The relationship can also exist between classes. For example: "car is a vehicle". Here, both 'car' and 'vehicle' are classes as there can be multiple types of both and this is an example of class-class relationship in hypernymy. Terms that have the same hypernym are called co-hyponyms. For example: Spain and France are co-hyponyms as they have the same hypernym, country.

The earliest attempts at detecting hypernym pairs from text started with the introduction of Hearst patterns (Hearst, 1992). This approach attempted to extract the hypernyms from the text using lexico-synctactic patterns that could capture the contexts in which hyponym-hypernym pairs occur in text. These patterns take advantage of noun phrases in a given corpus. Even though Hearst patterns may capture the hyponym-hypernym pairs from the corpus, they suffer from sparsity, that is, if the pairs do not follow the exact pattern that is used, then no relation is picked up.

Recent works are now moving to the distributional methods for hypernym detection which are based on the **DIH (distributional inclusion hypotheses)** (Geffet and Dagan, 2005), which states that the contexts in which a narrower term like 'Spain' occurs should be a subset of the contexts in which the broader term 'country' occurs. The measures in this space follow on from the creation of distributional semantic spaces and then use inclusion (Weeds et al., 2004) or non-inclusion (Lenci and Benotto, 2012) measures to detect if the hypernym relation holds. There is an alternative to the inclusion hypotheses, called the **informativeness** hypotheses, which uses entropy instead of inclusion contexts. This has been covered in Santus et al. (2014) and furthered in Shwartz et al. (2016b). Along with distributional approaches, there are some machine learning based approaches that introduce the idea of using dependency paths as features for known hypernym pairs (Snow et al., 2005) and further work branching out from this using satellite links (Sheena et al., 2016). Both referenced works train a classifier to predict whether the relation holds between two terms. There has also been work in the field of using distributional semantic spaces called embeddings (Mikolov et al., 2013; Pennington et al., 2014) to train classifiers for predicting hypernymy.

Recent works on hypernym detection have used Wikipedia derived corpora (Shwartz et al., 2016a) or Gigaword (Graff, David, and Christopher Cieri, 2011) concatenated with Wikipedia (Roller et al., 2018). Evaluation of extractions from these corpora has been done using 5 datasets which will be covered later in this paper (Section 3.2.3). Being consistent with Roller et al. (2018) and Shwartz et al. (2016b), average precision is used as a metric to evaluate extractions and predict hypernymy between pairs in all datasets.

In this paper, we first describe the two corpora from which our corpus is derived. We also detail the various approaches to hypernym detection and our methodology in extracting candidate pairs from the corpus. Finally, we describe the evaluation datasets used and compare our results to the current state-of-the-art (Roller et al., 2018).

We propose a free corpus along with its POS-tagged and dependency parsed versions that produces similar results on 5 tests and 4 methods. This is the main contribution of the paper [1] along with the relevant code for implementation [2], and the hyponym-hypernym pairs extracted.

2. Corpus Description

Our corpus has been created as a concatenation of two web-based sources that are provided not only in their raw format but also POS-tagged with dependency path annotations using spacy (Honnibal and Montani, 2017). Both sources are described in the following sections.

[1] DOI: 10.5281/zenodo.3662204
[2] https://github.com/abyssnlp/Hypernym-LIBre

2.1. UMBC Web Corpus

The UMBC [3] corpus (Han et al., 2013) is based on the Stanford WebBase crawl from February 2007 and contains 100 million web pages from 50,000 websites. Duplicated paragraphs and non-English words as well as strange characters were removed from the text to get 3 billion good quality English words. The corpus can be downloaded freely as a 13GB tar file which when uncompressed, comes to around 48GB of text + part-of-speech tagged files. There are 408 files which contain English text in the paragraph format and 408 files that are the same paragraphs but part-of-speech tagged.

2.2. Wikipedia Corpus

The English Wikipedia corpus is a widely used corpus in the field of Computational Linguistics and Natural Language Processing. It provides data for various fields of research as a one-stop online free encyclopedia. It also provides various APIs for extracting specific information and the entire Wikipedia in downloadable format [4] either in XML or SQL for directly integrating into a database for further analyses. Wikipedia as a corpus is especially useful in the field of Hypernym detection because it covers a variety of topics which can be extracted as candidate pairs for satisfying the relation.

2.3. Part-of-Speech Tagging and Dependency Parsing of our corpus

The UMBC corpus comes with 408 files of POS-tagged version of the their text counterparts which is almost 30GB. According to Han et al. (2013), the corpus was POS-tagged using the Stanford POS Tagger (Toutanova and Manning, 2000). As the POS-tagged version of UMBC is quite dated and we needed to POS-tag Wikipedia as well to extract noun-phrases from the corpora, we used the multi-task CNN(Convolutional Neural Network) from spacy (Honnibal and Montani, 2017) for the concatenation of both. Although we do not use dependency parsing in our models or experiments, it is useful for implementing some distributional models as listed in Shwartz et al. (2016b). We therefore, provide the dependency parsed version of the corpus as well for aiding future research in this field. This has also been performed using the dependency parser available in spacy.

3. Hypernym Detection

We analyze the state-of-the-art pattern-based methods for hypernym detection from Roller et al. (2018) and our evaluation shows that the results using our corpus are similar to the results from the alternate paid corpus mentioned before.

3.1. Approaches for Hypernym Detection

There are 3 main groups of approaches for hypernym detection that we enlist below.

[3] https://ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/

[4] https://dumps.wikimedia.org/

3.1.1. Pattern Based Methods

Pattern-based methods are the current state-of-the-art in Hypernym detection (Roller et al., 2018).
These methods use lexico-syntactic patterns (LSPs) to extract hypernym pairs based on their linguistic structure.
The most popular patterns were proposed by Hearst (1992), as shown in Table 1, where NP stands for noun phrases. Apart from the regular Hearst Patterns, more patterns can be used to extract hypernym-hyponyms from a corpus.

3.1.2. Unsupervised Distributional Methods

These methods involve the formation of distributional semantic spaces or DSMs to capture the contexts in which a word occurs. It is closely linked to how word embeddings like Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) are formed.
A vector space is created based on these contexts, and can be used to determine whether two words hold the hypernym relation.Vector spaces can be created using window-based approaches(taking a fixed window around the target word) or dependency-tree based(taking the parent and sister node of the target word in a dependency tree). For example, Let us consider a sentence: "Trade laws in **Uganda** are similar to those in South Africa." In this sentence, if we do not know what **Uganda** is, looking at the contexts surrounding this word and projecting it into a vector space of similar contexts, we can infer that it must be a country. A common method for checking for similarity in distributional spaces is Cosine Similarity (Dillon, 1983). After the creation of such a distributional semantic space, various measures can be applied for hypernymy detection. All measures are variants of the DIH (Distributional Inclusion Hypothesis) (Geffet and Dagan, 2005), which states that a narrower term's contexts will always be a subset of the broader term's contexts. For example: The context in which a narrower term like *dog* appears will be always be a subset of the contexts of a broader term like *animal*. All DIH measures are defined for large, sparse and positively valued distributional spaces. There are 3 main variants based on this:

- *WeedsPrec* (Weeds et al., 2004) which captures contexts of x that are included in the set of a broader term's contexts like y

$$WeedsPrec(x,y) = \frac{\sum_{i=1}^{n} x_i * \mathbb{1}_{y_i} > 0}{\sum_{i=1}^{n} x_i} \quad (1)$$

- *invCL* (Lenci and Benotto, 2012) which uses distributional inclusion as well as distributional exclusion of the contexts of the two words. It uses the inclusion variant from Clarke (2009) and adds a non-inclusion element to it.

$$CL(x,y) = \frac{\sum_{i=1}^{n} min(x_i, y_i}{\sum_{i=1} nx_i} \quad (2)$$

Hearst Patterns	
Pattern #1	NP_0 *such as* NP_1, NP_2 ..., $(and \mid or)$ NP_n Example: " Countries such as Spain, France and Germany." Extracts: NP_0: Countries (hypernym), NP_1: Spain (hyponym), NP_2: France (hyponym)
Pattern #2	*such* NP_0 *as* $\{NP_1, \} * \{(or \mid and)\}$ NP_n Example: "such flowers as Hibiscus and Rose." Extracts: NP_0: Flowers (hypernym), NP_1: Hibiscus (hyponym) and NP_2: Rose (hyponym)
Pattern #3	NP_0 $\{, NP_1\} * \{, \}$ *or other* NP_2 Example: "Enid Blyton, Mario Puzo or other authors." Extracts: NP_0: Enid Blyton (hyponym), NP_1: Mario Puzo (hyponym), NP_2: authors (hypernym)
Pattern #4	NP_0 $\{, NP_1\} * \{, \}$ *and other* NP_2 Example: "Socrates, Plato and other philosophers." Extracts: NP_0: Socrates (hyponym), NP_1: Plato (hyponym), NP_2: philosophers (hypernym)
Pattern #5	NP_0 $\{, \}$ *including* $\{NP_1, \} * \{or \mid and\} NP_2$ Example: "Fishes including Dolphins and Rays." Extracts: NP_0: Fishes (hypernym), NP_1: Dolphins (hyponym), NP_2: Rays (hyponym)
Pattern #6	NP_0 $\{, \}$ *especially* $\{NP_1, \} * \{or \mid and\} NP_2$ Example: "East European countries especially Bosnia and Hungary." Extracts: NP_0: East_European_countries (hypernym), NP_1: Bosnia (hyponym), NP_2: Hungary (hyponym)

Table 1: Hearst Patterns, Marti Hearst(1992).

$$invCL(x, y) = \sqrt{CL(x, y) * (1 - CL(y, x)} \quad (3)$$

- *SLQS* (Santus et al., 2014; Shwartz et al., 2016b) which is based on the alternate **informativeness** hypothesis. It depends on the median entropy of a term's top N contexts. Here N becomes the hyperparameter for the model.

$$E_x = median_{i=1}^{N}[H(c_i)] \quad (4)$$

, where H is the Shannon entropy. Then SLQS model is defined as the ratio of its application on both the terms in the pair:

$$SLQS(x, y) = 1 - \frac{E_x}{E_y} \quad (5)$$

3.1.3. Machine Learning Based Approaches

Supervised learning methods have been used to classify whether two words hold the hypernymy relation or not. Methods such as in Snow et al. (2005) and Sheena et al. (2016), create a training set with dependency paths between known hypernym-hyponym pairs as the features and the target as a binary variable whether that dependency path leads to a hypernymy relation or not. This task then becomes a binary classification task and can be used as a Hypernym classifier between a pair of words, given the dependency path that links them.

There has been recent progress in using neural networks and spherical embeddings (Wang et al., 2019) and in combining pattern-based approaches with nearest-neighbor candidate pairs (Held and Habash, 2019). However, these have not been considered in this study and are beyond the scope of this paper.

3.2. A Pattern-based Methodology for Hypernym Detection

Roller et al. (2018) conclude that pattern-based methods outperform distributional methods for Hypernym detection. In order to validate extractions, a corpus is required to match the patterns and obtain candidate hypernym-hyponym pairs. The dataset used in Roller et al. (2018) consisted of the concatenation of the Gigaword (Graff, David, and Christopher Cieri, 2011) and the Wikipedia corpus.

However, Gigaword is a paid corpus and requires fees for access. We used an alternate corpus derived from the concatenation of the UMBC and the Wikipedia corpus. A relevant result is that using our free corpus, we were able to achieve similar state-of-the-art results for all the datasets the extractions were validated on.

We now outline our methodology for obtaining these results using Pattern-based methods for Hypernym detection.

3.2.1. Extracting Pairs from the Corpus

Pairs were extracted from the UMBC+Wikipedia corpus as follows:

1. Convert the Hearst Patterns shown in Table 1 into regular expressions.
2. Pre-process and clean the corpus by removing special characters like #,$, HTML tags etc.
3. Split the corpus into sentences and tokenize each sentence into words such that we get a list of sentences where each sentence is a list of words.
4. Part-of-speech tag the words in each sentence with the Perceptron Tagger[5] [6].
5. Extract noun phrases from the text. Sequential noun phrases are combined into one with a single 'NP_' header
6. Match Hearst Patterns with the text and extract hyponym-hypernym pairs.

3.2.2. Matrix Operations on the Extractions

After extracting the pairs from the corpus, we compress them to get each unique pair and the frequency of extraction from the total extractions. This creates a counts table where we have the pair extracted alongside the number of times (frequency) of occurrence.

From these pairs and counts, we create a sparse co-occurrence matrix of all the words in the vocabulary where the rows are the hyponyms from each pair and the columns are the hypernyms. The value of each cell in the matrix is the number of extractions of that particular hyponym-hypernym pair or the frequency.

The *Raw Count Matrix* is created by dividing each value in the matrix by the total number of extractions to get the raw probability of extracting that particular pair as a valid hyponym-hypernym pair.

Let ρ denote the set of extractions from corpus τ ,

$$\rho = \{(x, y)\}_{i=1}^{n} \tag{6}$$

Let $w(x, y)$ denote the count of how often (x,y) has been extracted using our patterns from the corpus and the total number of extractions W be denoted as:

$$W = \sum_{(x,y)\in\rho} w(x, y) \tag{7}$$

In order to predict the hypernymy relation using this raw count matrix, we will use the probability of extraction of the pair as:

$$p(x, y) = \frac{w(x, y)}{W} \tag{8}$$

This is detailed in Algorithm 1.

However, raw count probabilities for predicting this relation suffers from word occurrence inconsistencies. For example, $(humans, mammals)$ are more likely to be extracted from the corpus than $(human, vertebrates)$, but both are true for hypernymy as humans are both mammals and vertebrates.

To deal with this, Roller et al. (2018) also used *PPMI (Positive Pointwise Mutual Information)* which is the mathematical translation of how likely are two words to occur together than occur independent of each other. We only take positive examples in this case as hypernymy is an asymmetric relationship. Although similarity is one of its properties, for example: blue is a color but the reverse is not true. As defined in Roller et al. (2018),

$$p^{-}(x) = \frac{\sum_{(x,y)\in\rho} w(x, y)}{W} \tag{9}$$

$$p^{+}(x) = \frac{\sum_{(y,x)\in\rho} w(y, x)}{W} \tag{10}$$

where, $p^{-}(x)$ and $p + (x)$ are the probability that x occurs as a hyponym and hypernym respectively.

Then the PPMI for the extracted pair (x,y) can be computed as:

$$ppmi(x, y) = max(0, log\frac{p(x, y)}{p^{-}(x)p^{+}(y)}) \tag{11}$$

The PPMI matrix is implemented on the raw count matrix as show in Algorithm 2.

While this can deal with skewed word occurrence probabilities, we still cannot handle out-of-vocabulary or unseen pairs. Therefore, we compute low-rank embeddings of the PPMI and the raw count matrix so that we can generalize to unseen or new pairs. Towards this, we use SVD or *Singular Value decomposition*, which is a kind of matrix factorization and reduces the matrix on the basis of the hyperparameter k which captures the number of singular values to retain and truncates all the rest. This leads to similar words having similar representations.

Given,

Let SVD of matrix M,

$$M = U \sum V^{T} \tag{12}$$

Then, Truncated SVD of M,

$$Trunc.SVD = u_x^{T} \sum_{r} v_y \tag{13}$$

in which all but the r largest singular values are set to 0.

In the experiments, we consider the SVD of both the raw count as well as the PPMI matrix. Implementation and procedure are detailed in Algorithm 3.

3.2.3. Evaluation Datasets

The 5 datasets used in the evaluation of our pattern-based methods are consistent with Roller et al. (2018) and Shwartz et al. (2016b).

Below we outline and detail the 5 datasets used:

[5] https://www.nltk.org/_modules/nltk/tag/perceptron.html

[6] Please note that while running the experiment, we POS tagged the corpus using the Perceptron Tagger. As the spacy tagger has been shown to be perform better, we POS tagged Hypernym-LIBre with it before releasing it as a language resource.

Algorithm 1 Raw Count Matrix from Hearst Patterns

$p \leftarrow (x, y)_{i=1}^{n}$ ▷ (x,y) - hyponym,hypernym pairs
$w(x, y) \leftarrow freq(x, y)$ ▷ frequency of extraction
$W \leftarrow \sum_{(x,y)\in p} w(x, y)$ ▷ total extractions
for $i := 1 \rightarrow n$ **do**
    $P(x_i, y_i) \leftarrow \frac{w(x_i,y_i)}{W}$
end for

Algorithm 2 PMI (Pointwise Mutual Information) on Raw Count Matrix

$p^{-}(x) \leftarrow \sum_{row} x$ ▷ prob(x as hyponym)
$p^{+}(x) \leftarrow \sum_{col} x$ ▷ prob(x as hypernym)
$p(x, y) \leftarrow \frac{w(x,y)}{W}$ ▷ from Algorithm 1
for $i := 1 \rightarrow n$ **do**
    $PMI(x_i, y_i) \leftarrow log\frac{p(x_i,y_i)}{p^{-}(x_i)\cdot p^{+}(y_i)}$
end for

1. BLESS (Baroni and Lenci, 2011)
   This dataset contains hypernymy annotations for around 200 nouns. It contains pairs for other relations like meronymy and co-hyponymy as well. We label the hypernym pairs as true and all other relations as false. It contains 14,542 total pairs with 1,337 positive examples.

2. LEDS (Baroni et al., 2012)
   This dataset consists of 2,770 nouns and comes balanced with randomly shuffled positive as well as negative pairs.

3. EVAL (Santus et al., 2015)
   This dataset contains 7,378 pairs in a mixture of hypernym ,antonym and synonym pairs. We only mark the hypernym pairs as true and all other relations as false.

4. SHWARTZ (Shwartz et al., 2016a)
   This is the largest dataset used. We took a subset containing 52,578 pairs (Roller et al., 2018).

5. WBLESS (Weeds et al., 2014)
   A dataset of 1,668 subset of the BLESS dataset containing negative pairs from other close relations to confirm the validity of our predictions.

Average precision is used as metric to score all the models in this paper to be consistent with Roller et al. (2018) and Shwartz et al. (2016b).

3.2.4. Setup and Hardware

The pairs were extracted, processed and evaluated on a server with 8 Intel Xeon cores and 64 GB of RAM .
None of the models have a hyper-parameter except for SVD based models, for which we selected k=100 for all. We also performed experiments with various other values of k={10,20,50,100,1000} but they have been omitted from the results for the sake of brevity.

3.2.5. Results and Comparison

Our evaluation shows that the results using our corpus are similar to the results from the alternate paid corpus mentioned before. Prior to evaluating, we trim all the extractions from our corpus that are less than 2 as it helps control the sparsity of our extractions. Truncated SVD on the PPMI models achieve highest scores overall. This is due to its matrix completion properties as similar words have similar representations. There are some slight variations in the results which stem from the difference in the corpus used and/or the pre-processing methodologies. However, these slight variations are not unidirectional as we perform slightly better in some datasets and slight worse in others. Overall, the results are similar as can be seen in Table 2. The metric used here is average precision. It summarizes the precision-recall curve with the weighted mean of precision at each threshold, with the increase in recall from the previous threshold used as the weight.

$$AP(AveragePrecision) = \sum_{n}(R_n - R_{n-1})P_n \quad (14)$$

, where R_n and P_n are recall and precision at the n^{th} threshold.
The comparison is as detailed below (the darker bars with suffix '_sota' represent the results from Roller et al. (2018) and the lighter bars with suffix '_libre' represent our results):

1. BLESS Dataset

   On the BLESS dataset, we perform similar to Roller et al. (2018). Here, SVD applied on the PPMI matrix achieves an Average Precision score of 0.71 as compared to 0.76. (as shown in Figure 1)

2. LEDS Dataset

   Similarly in LEDS, some of our models outperform Roller et al. (2018) and achieve exact scores on the highest performing SVD on PPMI matrix model. LEDS contains noun pairs which are discriminative and hence we get high scores overall. (as shown in Figure 2)

3. EVAL Dataset
   This dataset has some out-of-vocabulary words with

Algorithm 3 SVD (Singular Value Decomposition) on Raw Count and PPMI Matrix

$C \leftarrow raw\ count\ matrix/PPMI\ matrix$ ▷ from Algorithm 1/2
$k \leftarrow 100$ ▷ hyperparameter k
$r \leftarrow rank(C)$
$SVD(C) \leftarrow U \cdot \sum \cdot V^T$
$\sum_k \subset \sum$ ▷ truncated SVD by selecting k=100 singular values
$C_k \leftarrow U \cdot \sum_k \cdot V^T$ ▷ final matrix to use for predictions

Result Comparison								
Datasets	**Models**							
	Raw Count Model		**PPMI Model**		**SVD Raw Count Model**		**SVD PPMI Model**	
	SOTA	LIBre	SOTA	LIBre	SOTA	LIBre	SOTA	LIBre
BLESS	0.49	0.47	0.45	0.42	0.66	0.64	0.76	0.71
LEDS	0.71	0.73	0.7	0.73	0.81	0.82	0.84	0.84
EVAL	0.38	0.35	0.36	0.32	0.45	0.42	0.48	0.42
SHWARTZ	0.29	0.36	0.28	0.33	0.41	0.53	0.44	0.47
WBLESS	0.74	0.74	0.72	0.73	0.91	0.93	0.96	0.95

Table 2: Result Comparison between extractions from state-of-the-art corpus and Hypernym-LIBre.

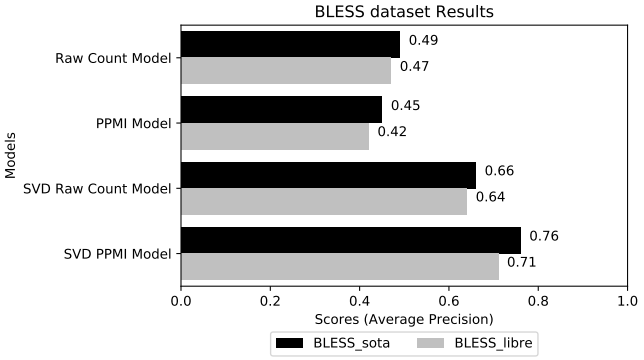

Figure 1: Pattern based methods on BLESS dataset

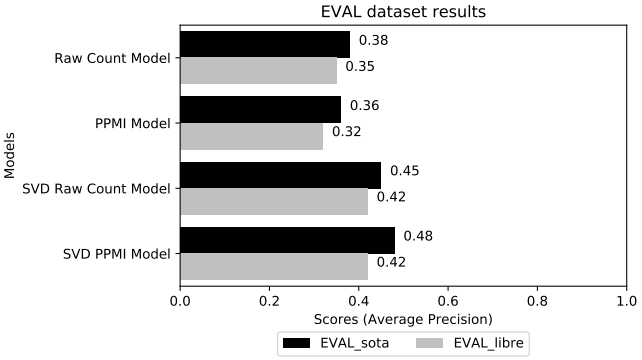

Figure 3: Pattern based methods on EVAL dataset

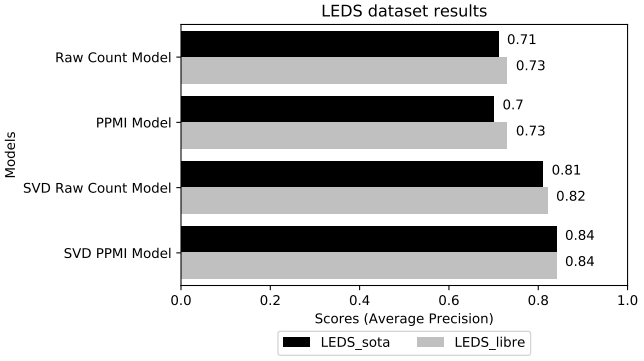

Figure 2: Pattern based methods on LEDS dataset

respect to our corpus from which we extracted our pairs and most of the pairs are verb or adjective pairs. Since our patterns extract noun pairs from the corpus, the score gets penalized by these pairs. Here we achieve 0.42 AP on the SVD PPMI model as compared to 0.48. (as shown in Figure 3)

4. SHWARTZ Dataset

This dataset is the largest dataset and it also has some

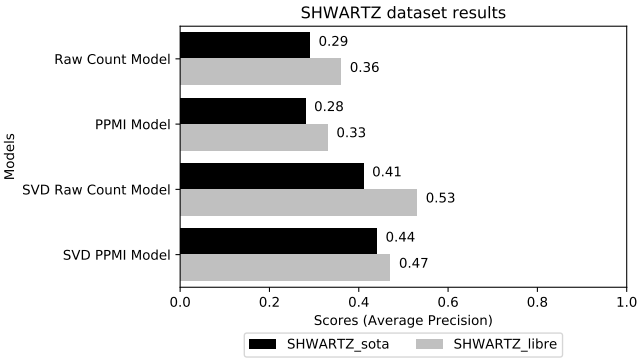

Figure 4: Pattern based methods on SHWARTZ dataset

very low frequency words which are not picked up by our Hearst Pattern based models and hence the overall score is low for all models. Here, we are at level with the state-of-the-art scores. (as shown in Figure 4)

5. WBLESS Dataset

This dataset scores very high on AP across all the

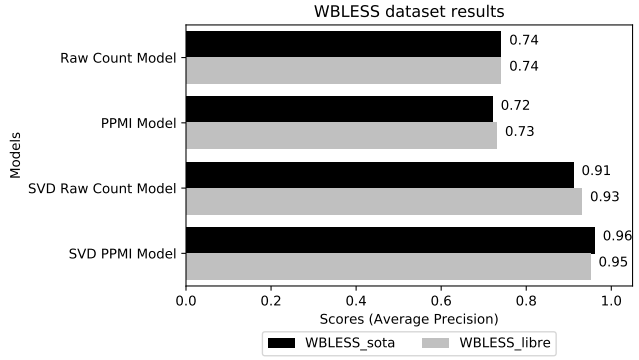

Figure 5: Pattern based methods on WBLESS dataset

models. Here, the SVD applied to PPMI matrix model achieves 0.95 AP compared to 0.96. (as shown in Figure 5)

4. Conclusion

We have created a new corpus that can be used by those working in methods and techniques for hypernym detection. Our evaluation shows that we get similar results when we apply state-of-the-art methods to it, hence showing that the corpus can be used for the same purpose as it has been done with previous corpora in the state-of-the-art, with the benefit of using a corpus that is available under an open license. In order to show that the usage of this corpus does not have a negative impact in comparison with the usage of previous ones, we also show how we applied all the pattern-based methods described in Roller et al. (2018) with our new corpus achieving similar results.

As future work, we plan to improve existing pattern-based methods using better or more patterns and generalization techniques. We also plan on testing the combination of distributional and pattern-based approaches.

5. Bibliographical References

Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.

Baroni, M., Bernardi, R., Do, N.-Q., and Shan, C.-c. (2012). Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32. Association for Computational Linguistics.

Clarke, D. (2009). Context-theoretic semantics for natural language: an overview. In *Proceedings of the workshop on geometrical models of natural language semantics*, pages 112–119.

Dillon, M. (1983). Introduction to modern information retrieval: G. salton and m. mcgill. mcgraw-hill, new york (1983). xv+ 448 pp., 32.95 isbn 0-07-054484-0.

Geffet, M. and Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 107–114. Association for Computational Linguistics.

Han, L., Kashyap, A. L., Finin, T., Mayfield, J., and Weese, J. (2013). Umbc_ebiquity-core: Semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 44–52.

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

Held, W. and Habash, N. (2019). The effectiveness of simple hybrid systems for hypernym discovery. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3362–3367.

Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Lenci, A. and Benotto, G. (2012). Identifying hypernyms in distributional semantic spaces. In * *SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Roller, S., Kiela, D., and Nickel, M. (2018). Hearst patterns revisited: Automatic hypernym detection from large text corpora. *arXiv preprint arXiv:1806.03191*.

Santus, E., Lenci, A., Lu, Q., and Im Walde, S. S. (2014). Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42.

Santus, E., Yung, F., Lenci, A., and Huang, C.-R. (2015). Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69.

Sheena, N., Jasmine, S. M., and Joseph, S. (2016). Automatic extraction of hypernym & meronym relations in english sentences using dependency parser. *Procedia Computer Science*, 93:539–546.

Shwartz, V., Goldberg, Y., and Dagan, I. (2016a). Improving hypernymy detection with an integrated

path-based and distributional method. *arXiv preprint arXiv:1603.06076*.

Shwartz, V., Santus, E., and Schlechtweg, D. (2016b). Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. *arXiv preprint arXiv:1612.04460*.

Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems*, pages 1297–1304.

Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics.

Wang, C., He, X., and Zhou, A. (2019). Spherere: Distinguishing lexical relations with hyperspherical relation embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1727–1737.

Weeds, J., Weir, D., and McCarthy, D. (2004). Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1015. Association for Computational Linguistics.

Weeds, J., Clarke, D., Reffin, J., Weir, D., 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 City University and Association for Computational Linguistics.

6. Language Resource References

Graff, David, and Christopher Cieri. (2011). *English Gigaword*. Linguistic Data Consortium, 5.0, ISLRN 953-543-425-922-6.