Wordnet extension via word embeddings: Experiments on the Norwegian Wordnet

Heidi Sand and **Erik Velldal** and **Lilja Øvrelid**
University of Oslo
Department of Informatics
{heidispe,erikve,liljao}@ifi.uio.no

Abstract

This paper describes the process of automatically adding synsets and hypernymy relations to an existing wordnet based on word embeddings computed for POS-tagged lemmas in a large news corpus, achieving exact match attachment accuracy of over 80%. The reported experiments are based on the Norwegian Wordnet, but the method is language independent and also applicable to other wordnets. Moreover, this study also represents the first documented experiments of the Norwegian Wordnet.

1 Introduction

This paper documents experiments with an unsupervised method for extending a wordnet with new words and automatically identifying the appropriate hypernymy relations. Using word embeddings trained on a large corpus of news text (~330 million tokens), candidate hypernyms for a given target word are identified by computing nearest neighbors lists towards the wordnet and retrieving the ancestors of the neighbors. The candidate hypernyms are then scored according to a combination of distributional similarity and distance in the wordnet graph. While the particular experimental results reported here are for the Norwegian Wordnet (NWN), and using vectors estimated on POS tagged lemmas of the Norwegian news corpus, the methodology is generally applicable and not specific to neither the language nor the particular language resources used.

2 Background

Due to the coverage limitations of manually constructed semantic resources, several approaches have attempted to enrich various taxonomies with new relations and concepts. The general approach is to attempt to insert missing concepts into a taxonomy based on distributional evidence. In their probabilistic formulation, Snow et al. (2006) maximize the conditional probability of hyponym-hypernym relations based on observed lexico-syntactic patterns in a corpus. Jurgens and Pilehvar (2015) extend the existing WordNet taxonomy using an additional resource, Wiktionary, to extract sense data based on information (morphology/lexical overlap) in the term glosses.

The work which is most directly relevant to our study is that of Yamada et al. (2009). They extend an automatically generated Wikipedia-taxonomy by inserting new terms based on various similarity measures calculated from additional web documents. As the approach described in the current paper will abstractly adapt the approach of Yamada et al. (2009), we will devote some space to elaborate how the algorithm works, glossing over details that does not pertain to our setting. The insertion mechanism works as follows: For a given target word w we first find the k similar words that are already present in the hierarchy, according to some distributional similarity measure *sim* and with the constraint that the similarity is greater than some cutoff m. Secondly the hypernyms of each of these k similar words are assigned scores, based on a combination of the similarity measure and a depth penalty d. The latter is a function of the distance r in the hierarchy between the neighbor and the hypernym, as the hypernym candidates also include ancestor nodes beyond the immediate parents. Finally, the hypernym $h_\uparrow$ with the highest score will be selected for attaching w.

The function used for scoring hypernym candidates is defined as follows (paraphrased here according to the notation of the current paper):

$$score(h_\uparrow) = \sum_{h_\downarrow \in desc(h_\uparrow) \cap ksim(w)} d^{r(h_\uparrow, h_\downarrow)-1} \times sim(w, h_\downarrow) \qquad (1)$$

Proceedings of the 21st Nordic Conference of Computational Linguistics, pages 298–302,
Gothenburg, Sweden, 23-24 May 2017.

$ksim(w)$ picks out the k nearest neighbors of the target word w according to the distributional similarity measure *sim*, with the constraint that the similarity is greater than a cutoff m. The function $desc(h_\uparrow)$ picks out the descendants (hyponyms) of the candidate hypernym $h_\uparrow$. The term $d^{r(h_\uparrow,h_\downarrow)-1}$ is the depth penalty where the parameter d can have a value between 0 and 1, $r(n_\uparrow, n_\downarrow)$ is the difference in hierarchy depth between $h_\uparrow$ and $h_\downarrow$.

For *sim*, two distributional similarity measures are tested by Yamada et al. (2009), based on respectively 1) raw verb-noun dependencies and 2) clustering of verb-noun dependencies. Yamada et al. (2009) also apply a baseline approach, selecting the hypernym of the most similar hyponym (baseline approach 1), which essentially is the same as specifying k=1 when computing the nearest neighbors in the approach outlined above. Manually evaluating a random sample of 200 of the highest scoring insertions, Yamada et al. (2009) report an attachment accuracy of up to 91.0% among the top 10,000, and 74.5% among the top 100,000, when using the clustering based similarity – a result which substantially improves over the baseline (yielding scores of ~55%).

Although Yamada et al. (2009) work with a semantic taxonomy based on the Wikipedia structure, the scoring function in Equation 1 which forms the pivot of the approach is general enough to be adopted for other settings as well. Notably, no assumptions are made about the particular similarity function instantiating $sim(w_i, w_j)$. In the work reported in the current paper we experiment with instead using a similarity function based on word embeddings computed from a large unannotated corpus, and apply this for extending NWN.

3 The Norwegian Wordnet

The Norwegian Wordnet (NWN) was created by translation from the Danish Wordnet (DanNet) (Pedersen et al., 2009). DanNet encodes both relations found in Princeton Wordnet (Fellbaum, 1998) and EuroWordNet (Vossen, 1998), some of which are also employed in NWN. There are no prior publications documenting NWN, and the current study provides the first reported experiments on this resource. Before commencing our experiments it was therefore necessary to pre-process NWN in order to (i) correct errors in the format and/or the structure of NWN, and (ii) remove named entities and multiword expressions. The resulting wordnet is summarized in Table 1, showing the number of lemmas, synsets and monosemous/polysemous terms broken down by their part-of-speech (nouns, adjectives and verbs). The modified NWN, which forms the basis for our experiments, is made freely available.[1]

POS	Lemmas	Synsets	Senses	Monos.	Polys.
Noun	38,440	43,112	48,865	31,957	6,483
Verb	2,816	4,967	5,580	1,612	1,204
Adj	2,877	3,179	3,571	2,413	464
Total	44,133	51,258	58,016	35,982	8,151

Table 1: Number of lemmas, synsets, senses and monosemous/polysemous words for nouns, verbs and adjectives in NWN.

4 Word embeddings for tagged lemmas

This section describes how we generate the semantic context vectors representing both unseen target words and the words already present in the existing wordnet synsets. These vectors form the basis of our distributional similarity measure, used both for computing nearest neighbors within NWN for unclassified target words and for scoring candidate hypernyms according to Equation 1 (where the similarity function will correspond to the cosine of word vectors). Our semantic vectors are given as word2vec-based word embeddings (Mikolov et al., 2013a; Mikolov et al., 2013b) estimated on the Norwegian Newspaper Corpus[2]. This is a corpus of Norwegian newspaper texts from the time period 1998–2014. We used approximately 25% of this corpus (due to some technical issues), which amounts to 331,752,921 tokens and 3,014,704 types.

Rather than estimating word embeddings from raw text, we use POS-tagged lemmas in order to have embeddings that more closely correspond to the word representations found in NWN. In order to extract lemmas and their parts of speech, we pre-processed the data using the Oslo-Bergen Tagger[3] (Johannessen et al., 2012), a rule-based POS-tagger for Norwegian which also performs tokenization and lemmatization. The tagger was accessed through the Language Analysis Portal (LAP[4]) which provides a graphical web interface

[1] https://github.com/heisand/NWN
[2] http://www.nb.no/sprakbanken/repositorium
[3] http://www.tekstlab.uio.no/obt-ny
[4] https://lap.hpc.uio.no

to a wide range of language technology tools (Lapponi et al., 2013).

Word2vec implements two model types: continuous bag-of-words (CBOW) and skip-gram. These differ in the prediction task they are trained to solve: prediction of target words given context words (CBOW) or the inverse, prediction of context words given target words (skip-gram). We used the word2vec implementation provided in the free python library *gensim* (Řehůřek and Sojka, 2010), using the default parameters to train skip-gram models. The defaults are a minimum of 5 occurrences in the corpus for the lemmas and an embedding dimension of size 100. Five iterations over the corpus was made.

5 The attachment process

In this section we detail the steps involved in classifying a new word w in the wordnet hierarchy.

5.1 Selecting nearest neighbors

The first step in the process is to compute the list of k nearest neighbors of w according to the distributional similarity $sim(w, w')$, a measure which in our case corresponds to the cosine of word embeddings described in Section 4. Candidate neighbors are words that are *(a)* already defined in NWN, *(b)* have a hypernym in NWN, *(c)* have the same part of speech as the target and *(d)* occur in the news corpus with a sufficient frequency to have a word embedding in the model described in Section 4. In addition we discard any neighbors that have a similarity less than some specified threshold m. As described in Section 6 we tune both k and m empirically.

5.2 Selecting candidate hypernyms

Hypernymy is a relation between *synsets* representing word senses, which means that the process of selecting candidate hypernyms for scoring has the following steps: First we *(a)* identify the list of k nearest neighbors for a given target word w, and then *(b)* for each neighbor word retrieve all synsets that encode a sense for that word, before we finally *(c)* retrieve all hypernym synsets, including all ancestor nodes, for those synsets.

Each candidate hypernym synset $h_\uparrow$ will in turn be assigned a score according to Equation 1. The synset with the highest score is finally chosen as the hypernym synset for the target word to be attached in NWN. Note that a given target word will only be assigned a single hypernym.

5.3 Evaluation

There are several ways one could choose to evaluate the quality of the words that are automatically inserted into the hierarchy. For example, Yamada et al. (2009) chose to manually evaluate a random sample of 200 unseen words, while Jurgens and Pilehvar (2015) treat the words already encoded in the hierarchy as gold data and then try to reattach these. We here follow the latter approach. However, while Jurgens and Pilehvar (2015) restrict their evaluation to monosemous words, we also include polysemous words in order to make the evaluation more realistic.

For evaluation and tuning we split the wordnet into a development set and a test set, with 1388 target words in each. Potential targets only comprise words that have a hypernym encoded (which, in fact, are not that many, as NWN is relatively flat) and occur in the news corpus sufficiently often (≥ 5) to be represented by a word embedding.

We evaluate hypernym selection according to both *accuracy* and *attachment*. While accuracy reflects the percentage of target words added that were correctly placed under the right hypernym, the attachment score is the percentage of target words that actually were inserted into NWN. A candidate target word might end up not getting attached if it has no neighbors fulfilling the requirements described in Section 5.1.

Computing accuracy based only on exactly correct insertions is rather strict. Intuitively, a hypernymy relation can be right or wrong with varying degrees. We therefore also include a *soft accuracy* measure that aims to take account of this by counting how many hyponym or hypernym edges that separates a lemma from its correct position. Each edge will weight the count by a factor of 0.5, partly based on the accuracy measure of Jurgens and Pilehvar (2015), who, instead of weighting the score, only measures accuracy as the number of edges away that a lemma is placed from its original position. We defined the formula for *soft accuracy* as:

$$\frac{count(correct) + \sum_0^{count(misplaced)} 1 * 0.5^{edges}}{count(attached)} \quad (2)$$

6 Experiments and results

The parameters that need to be empirically tuned are: the depth penalty d, the number of k nearest

≥Freq.	Dev. set	#Words	Att.	Acc.	Soft
5	1388	1337	96.33	55.80	63.25
100	854	840	98.36	61.67	68.08
500	461	448	97.18	62.50	68.89
1000	316	304	96.20	64.47	70.85

Table 2: Accuracy restricted to target words with a frequency higher than some given threshold. #Words shows the number of attached words.

neighbors to consider, and the minimum threshold m for the similarity of neighbors towards the target. After an initial round of experiments that determined the approximate appropriate range of values for these parameters, we performed an exhaustive grid search for the best parameter combination among the following values:

$k \in [1, 12]$ in increments of 1.

$m \in [0.5, 0.9]$ in increments of 0.05.

$d \in [0.05, 0.5]$ in increments of 0.05.

Optimizing for attachment accuracy, the best parameter configuration after tuning on the development set was found to be k=6, m=0.5, and d=0.05, yielding an accuracy of 55.80% and a degree of attachment of 96.33%.

As one might expect that the embeddings are more reliable for high-frequent words than for low-frequent words, we also computed the dev-set accuracy relative to frequency of occurrence in the corpus used for estimating the embeddings. The results are shown in Table 2. We indeed see that the accuracy goes up when enforcing a higher frequency cutoff, reaching 64.47% when setting the cutoff to 1000, though per definition this means sacrificing coverage.

We also evaluated the effect of only inserting words that had hypernyms with a score higher than a given cutoff, which again naturally leads to a lower degree of attachment. Table 3 shows the accuracies over the development set when enforcing different cutoffs, showing an increased accuracy. We see that the best performance is when the cutoff on the hypernym score is set to 4.6, with a corresponding attachment accuracy of 83.26%.

Held-out results Applying the model configuration (without cutoffs) to the held-out test words of NWN yields an attachment of 95.97% and an accuracy of 59.91% (soft acc. = 66.04%). We see that there is a slight increase in the accuracy

≥Hyp. score	#Words	Att.	Acc.	Soft
0.2	1337	96.33	55.80	63.25
1.0	1185	85.38	59.41	66.85
1.8	958	69.02	65.66	73.20
2.6	720	51.87	72.92	80.16
3.4	505	36.38	78.22	85.00
4.6	239	17.22	83.26	89.33

Table 3: Accuracy restricted to hypernyms with a score higher than some given threshold, computed over the 1388 words in the development set. #Words shows the number of attached words, e.g. 1337 is 96.33% of 1388.

for the insertions performed with the word embeddings when moving from the development data to the held-out data. As a baseline approach we also tried attaching each target word to the hypernym of its 1-nearest-neighbor. (When there are several candidate hypernyms available, we simply pick the first candidate in the retrieved list.) Yielding an accuracy of 47.61%, it is clear that we improve substantially over the baseline when instead performing insertion using the scoring function.

Applying the scoring function to the test set using the cutoff with the highest accuracy from Table 3, yields an accuracy of 84.96% (soft = 90.38%), though at the cost of a lower attachment rate (16.28%).

7 Summary and further work

This paper has demonstrated the feasibility of using word embeddings for automatically extending a wordnet with new words and assigning hypernym relations to them. When scoring candidate hypernyms we adopt the scoring function of Yamada et al. (2009) and show that this yields high accuracy even-though we apply it with a different type of taxonomic hierarchy and different types of distributional similarity measures. We compute distributional similarity based on word embeddings estimated from the Norwegian news corpus, using this as our basis for automatically attaching new words into hypernym relations in the Norwegian Wordnet, with exact-match accuracies of over 80%. For immediate follow-up work we plan to let the parameter tuning be optimized towards a combination of attachment and accuracy, rather than just accuracy alone.

References

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Janne Bondi Johannessen, Kristin Hagen, André Lynum, and Anders Nøklestad. 2012. Obt+stat: A combined rule-based and statistical tagger. In Gisle Andersen, editor, *Exploring Newspaper Language: Using the web to create and investigate a large corpus of modern Norwegian*. John Benjamins, Amsterdam, The Netherlands.

David Jurgens and Mohammad Taher Pilehvar. 2015. Reserating the awesometastic: An automatic extension of the wordnet taxonomy for novel terms. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT 2015)*.

Emanuele Lapponi, Erik Velldal, Nikolay Aleksandrov Vazov, and Stephan Oepen. 2013. HPC-ready language analysis for human beings. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*.

Bolette Sandford Pedersen, Sanni Nimb, Jørg Asmussen, Nicolai Hartvig Srensen, Lars Trap-Jensen, and Henrik Lorentzen. 2009. DanNet: the challenge of compiling a WordNet for Danish by reusing a monolingual dictionary. *Language Resources and Evaluation*, 43:269–299.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*.

Piek Vossen, editor. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer, Dordrecht, The Netherlands.

Ichiro Yamada, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, Masaki Murata, Stijn De Saeger, Francis Bond, and Asuka Sumida. 2009. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.