Two experiments for embedding Wordnet hierarchy into vector spaces*

Jean-Philippe Bernardy and Aleksandre Maskharashvili
Gothenburg University, Department of philosophy, linguistics and theory of science,
Centre for linguistics and studies in probability

jean-philippe.bernardy,aleksandre.maskharashvili@gu.se

Abstract

In this paper, we investigate mapping of the WORDNET hyponymy relation to feature vectors. Our aim is to model lexical knowledge in such a way that it can be used as input in generic machine-learning models, such as phrase entailment predictors. We propose two models. The first one leverages an existing mapping of words to feature vectors (*fast*Text), and attempts to classify such vectors as within or outside of each class. The second model is fully supervised, using solely WORDNET as a ground truth. It maps each concept to an interval or a disjunction thereof. The first model approaches but not quite attain state of the art performance. The second model can achieve near-perfect accuracy.

1 Introduction

Distributional encoding of word meanings from large corpora (Mikolov et al., 2013; Mikolov et al., 2018; Pennington et al., 2014) have been found to be useful for a number of NLP tasks.

While the major goal of distributional approaches is to identify distributional patterns of words and word sequences, they have even found use in tasks that require modeling more fine-grained relations between words than co-occurrence in word sequences. But distributional word embeddings are not easy to map onto ontological relations or *vice-versa*. We consider in this paper the hyponymy relation, also called the *is-a* relation, which is one of the most fundamental ontological relations. We take as the source of truth for hyponymy WORDNET (Fellbaum, 1998), which has been designed to include various kinds of lexical relations between words, phrases, etc. However, WORDNET has a fundamentally symbolic representation, which cannot be readily used as input to neural NLP models.

Several authors have proposed to encode hyponymy relations in feature vectors (Vilnis and McCallum, 2014; Vendrov et al., 2015; Athiwaratkun and Wilson, 2018; Nickel and Kiela, 2017). However, there does not seem to be a common consensus on the underlying properties of such encodings. In this paper, we aim to fill this gap and clearly characterize the properties that such an embedding should have. We additionally propose two baseline models approaching these properties: a simple mapping of FASTTEXT embeddings to the WORDNET hyponymy relation, and a (fully supervised) encoding of this relation in feature vectors.

2 Goals

We want to model the hyponymy relation (ground truth) given by WORDNET — hereafter referred to as HYPONYMY. In this section we make this goal precise and formal. Hyponymy can in general relate common noun phrases, verb phrases or any predicative phrase, but hereafter we abstract from all this and simply write "word" for this underlying set. In this paper, we write $(\subseteq)$ for the reflexive transitive closure of the hyponymy relation (ground truth), and $(\subseteq_M)$ for relation predicted by a model M.[1] Ideally, we want the model to be sound and complete with respect to the ground truth. However, a machine-learned model will typically only approach those properties to a certain level, so the usual relaxations are made:

Property 1 *(Partial soundness) A model M is*

*Supported by Swedish Research Council, Grant number 2014-39.

[1] We note right away that, on its own, the popular metric of cosine similarity (or indeed any metric) is incapable of modeling HYPONYMY, because it is an asymmetric relation. That is to say, we may know that the embedding of "animal" is close to that of "bird", but from that property we have no idea if we should conclude that "a bird is an animal" or rather that "an animal is a bird".

partially sound with precision α iff., for a proportion α of the pairs of words w, w' such that $w \subseteq_M w'$ holds, $w \subseteq w'$ holds as well.

Property 2 *(Partial completeness) A model M is partially complete with recall α iff., for a proportion α of the pairs of words w, w' such that $w \subseteq w'$ holds, then $w \subseteq_M w'$ holds as well.*

These properties do not constrain the way the relation $(\subseteq_M)$ is generated from a feature space. However, a satisfying way to generate the inclusion relation is by associating a subset of the vector space to each predicate, and leverage the inclusion from the feature space. Concretely, the mapping of words to subsets is done by a function P such that, given a word w and a feature vector x, $P(w, x)$ indicates if the word w applies to a situation (state of the world, sentence meaning, sentory input, etc.) described by feature vector x. We will refer to P as a classifier. The inclusion model is then fully characterized by P, so we can denote it as such $(\subseteq_P)$.

Property 3 *(Space-inclusion compatibility) There exists $P : (Word \times \mathbb{R}^d) \to [0, 1]$ such that*

$$(w' \subseteq_P w) \iff (\forall x. P(w, x) \leq P(w', x))$$

Any model given by such a P yields a relation $(\subseteq_P)$ which is necessarily reflexive and transitive (because subset inclusion is such) — the model does not have to learn this. Again, the above property will apply only to ideal situations: it needs to be relaxed in some machine-learning contexts. To this effect, we can define the measure of the subset of situations which satisfies a predicate $p : \mathbb{R}^d \to [0, 1]$ as follows:

$$\mathsf{measure}(p) = \int_{\mathbb{R}^d} p(x)dx$$

(Note that this is well-defined only if p is a measurable function over the measurable space of feature vectors.) We leave implicit the density of the vector space in this definition. Following this definition, a predicate p is included in a predicate q iff.

$$\frac{\mathsf{measure}(p \wedge q)}{\mathsf{measure}(p)} = \frac{\int_{\mathbb{R}^d} p(x)q(x)dx}{\int_{\mathbb{R}^d} p(x)dx} = 1$$

Following this thread, we can define a relaxed inclusion relation, corresponding to a proportion of ρ of p included in q:

Property 4 *(Relaxed Space-inclusion compatibility) There exists $P : Word \to \mathbb{R}^d \to [0, 1]$ and $\rho \in [0, 1]$ such that*

$$(w' \subseteq_P w) \iff \frac{\int_{\mathbb{R}^d} P(w', x)P(w, x)dx}{\int_{\mathbb{R}^d} P(w, x)dx} \geq \rho$$

In the following, we call ρ the relaxation factor.

3 Mapping WordNet over *fast*Text

Our first model of HYPONYMY works by leveraging a general-purpose, unsupervised method of generating word vectors. We use *fast*Text (Mikolov et al., 2018) as a modern representative of word-vector embeddings. Precisely, we use pre-trained word embeddings available on the *fast*Text webpage, trained on Wikipedia 2017 and the UMBC webbase corpus and the statmt.org news dataset (16B tokens). We call FTDom the set of words in these pre-trained embeddings.

A stepping stone towards modeling the inclusion relation correctly is modeling correctly each predicate individually. That is, we want to learn a separation between *fast*Text embeddings of words that belong to a given class (according to WordNet) from the words that do not. We let each word w in *fast*Text represent a situation corresponding to its word embedding $f(w)$. Formally, we aim to find P such that

Property 5 $P(w, f(w')) = 1 \iff w' \subseteq w$

for every word w and w' found both in WordNet and in the pre-trained embeddings. If the above property is always satisfied, the model is sound and complete, and satisfies Property 3.

Because many classes have few representative elements relative to the number of dimensions of the *fast*Text embeddings, we limit ourselves to a linear model for P, to limit the possibility of overfitting. That is, for any word w, $P(w)$ is entirely determined by a bias $b(w)$ and a vector $\theta(w)$ (with 300 dimensions):

$$P(w, x) = \delta(\theta(w) \cdot x + b(w) > 0)$$

where $\delta(\text{true}) = 1$ and $\delta(\text{false}) = 0$.

We learn $\theta(w)$ and $b(w)$ by using logistic regression, independently for each WordNet word w. The set of all positive examples for w is $\{f(w') \mid w' \in \mathsf{FTDom}, w' \subseteq w\}$, while the set of negative examples is $\{f(w') \mid w' \in \mathsf{FTDom}, w' \not\subseteq w\}$. We train and test for all the predicates with at least 10 positive examples. We

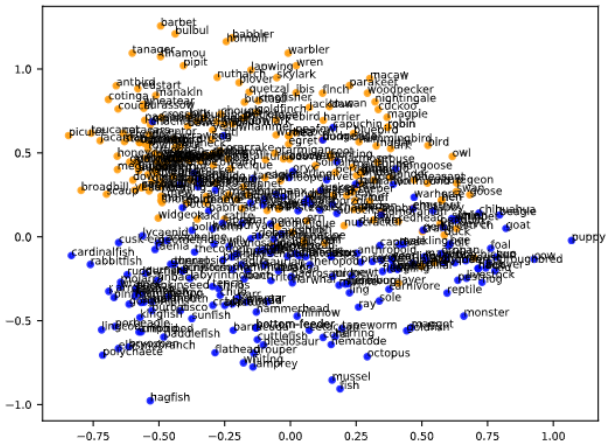

Figure 1: PCA representation of animals. Birds are highlighted in orange.

use 90% of the set of positive examples (w') for training (reserving 10% for testing) and we use the same number of negative examples.

We then test Property 5 on the 10% of positive examples reserved for testing, for each word. On average, we find that 89.4% of positives are identified correctly (std. dev. 14.6 points). On 1000 randomly selected negative examples, we find that on averale 89.7% are correctly classified (std dev. 5.9 points). The result for positives may look high, but because the number of true negative cases is typically much higher than that of true positives (often by a factor of 100), this means that the recall and precision are in fact very low for this task. That is, the classifier can often identify correctly a *random* situation, but this is a relatively easy task. Consider for example the predicate for "bird". If we test random negative entities ("democracy", "paper", "hour", etc.), then we may get more than 97% accuracy. However, if we pick our samples in a direct subclass, such as (non-bird) animals, we typically get only 75% accuracy. That is to say, 25% of animals are incorrectly classified as birds.

To get a better intuition for this result, we show a Principal Component Analysis (PCA) on animals, separating bird from non-birds. It shows mixing of the two classes. This mixture can be explained by the presence of many uncommon words in the database (e.g. types of birds that are only known to ornithologists). One might argue that we should not take such words into account. But this would severely limit the number of examples: there would be few classes where logistic regression would make sense.

We are not ready to admit defeat yet as we are ultimately not interested in Property 5, but rather in properties 1 and 2, which we address in the next section.

4 Inclusion of subsets

A strict interpretation of Property 3 would dictate to check if the subsets defined in the previous section are included in each other or not. However, there are several problems with this approach. To begin, hyperplanes defined by θ and b will (stochastically) always intersect therefore one must take into account the actual density of the *fast*Text embeddings. One possible approximation would be that they are within a ball of certain radius around the origin. However, this assumption is incorrect: modeling the density is a hard problem in itself. In fact, the density of word vectors is so low (due to the high dimensionality of the space) that the question may not make sense. Therefore, we refrain from making any conclusion on the inclusion relation of the subsets, and fall back to a more experimental approach.

Thus, we will test the suitability of the learned $P(w)$ by testing whether elements of its subclasses are contained in the superclass. That is, we define the following quantity $Q(w', w) =$

$$average\{P(w', x) \mid x \in \mathsf{FTDom}, P(w, f(x))\}$$

which is the proportion of elements of w' that are found in w. This value corresponds to the relaxation parameter ρ in Property 4.

If $w' \subseteq w$ holds, then we want $Q(w', w)$ to be close to 1, and close to 0 if w' is disjoint from w. We plot (figure 2) the distribution of $Q(w', w)$ for all pairs $w' \subseteq w$, and a random selection of pairs such that $w' \not\subseteq w$. The negative pairs are generated by taking all pairs (w', w) such that $w' \subseteq w$, and generate two pairs (w_1, w) and (w', w_2), by picking w_1 and w_2 at random, such that neither of the generated pairs is in the HYPONYMY relation. We see that most of the density is concentrated at the extrema. Thus, the exact choice of ρ has little influence on accuracy for the model. For $\rho = 0.5$, the recall is 88.8%. The ratio of false positives to the total number of negative test cases is 85.7%. However, we have a very large number of negatives cases (the square of the number of classes, about 7 billions). Because of this, we get about 1 billion false positives, and the precision is only 0.07%. Regardless, the results are comparable with state-of-the art models (section 6).

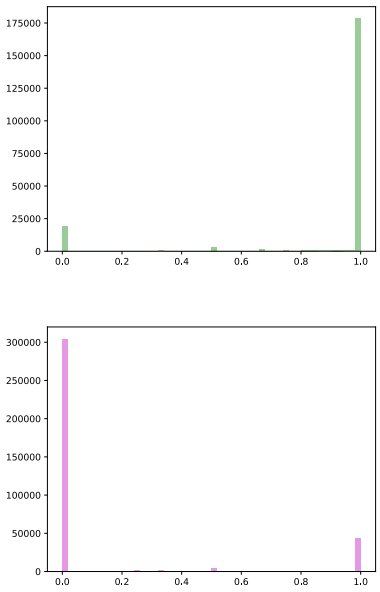

Figure 2: Results of inclusion tests. On the left-hand-side, we show the distribution of correctly identified inclusion relations in function of ρ. On the right-hand-side, we show the distribution of (incorrectly) identified inclusion relations in function of ρ.

5 WordNet predicates as disjunction of intervals

In this section we propose a baseline, fully supervised model model for HYPONYMY.

The key observation is that most of the HYPONYMY relation fits in a tree. Indeed, out of 82115 nouns, 7726 have no hypernym, 72967 have a single hypernym, and 1422 have two hypernyms or more. In fact, by removing only 1461 direct edges, we obtain a tree. The number of edges removed in the transitive closure of the relation varies, depending on which exact edges are removed, but a typical number is 10% of the edges. In other words, when removing edges in such a way, one lowers the recall to about 90%, but the precision remains 100%. Indeed, no pair is added to the HYPONYMY relation. This tree can then be mapped to one-dimensional intervals, by assigning a position to each of the nodes, according to their index in depth-first order ($ix(w)$ below). Then, each node is assigned an interval corresponding to the minimum and the maximum position assigned to their leaves. A possible directed acyclic graph (DAG) and a corresponding assignment of intervals is shown in Fig. 3. The corresponding definition of predicates is the following:

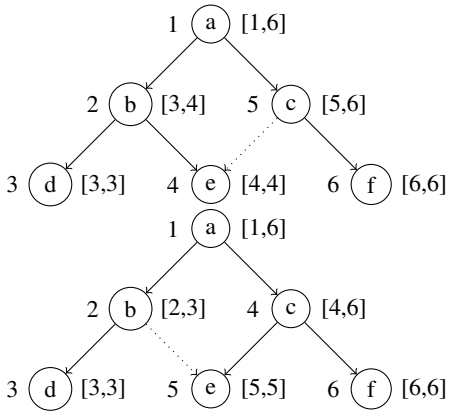

Figure 3: Two trees underlying the same dag. Nodes are labeled with their depth-first index on the left and their associated interval on the right. Removed edges are drawn as a dotted line.

$$\begin{aligned} P(w,x) &= x \geq lo(w) \wedge x \leq hi(w) \\ lo(w) &= min\{ix_T(w') \mid w' \subseteq_T w\} \\ hi(w) &= max\{ix_T(w') \mid w' \subseteq_T w\} \end{aligned}$$

where $(\subseteq_T)$ is the reflexive-transitive closure of the T tree relation (included in HYPONYMY). The pair of numbers $(lo(w), hi(w))$ fully characterizes $P(w)$. In other words, the above model is fully sound (precision=1), and has a recall of about 0.9. Additionally, Property 3 is verified.

Because it is fully sound, a model like the above can always be combined with another model to improve its recall with no impact on precision — including itself. Such a self-combination is useful if one does another choice of removed edges. Thus, each word is characterized by an n-dimensional co-product (disjoint sum) of intervals.

$$w \subseteq_M w' \triangleq \bigvee_i \Big(lo_i(w') \geq lo_i(w) \wedge hi_i(w') \leq hi_i(w) \Big)$$

$$lo_i(w) = min\{ix_{T_i}(w') \mid w' \subseteq_{T_i} w\}$$
$$hi_i(w) = max\{ix_{T_i}(w') \mid w' \subseteq_{T_i} w\}$$

By increasing n, one can increase the recall to obtain a near perfect model. Table 4b shows typical recall results for various values of n. However Property 3 is not verified: the co-product of intervals do not form subspaces in any measurable set.

6 Related Work: Precision and recall for hyponymy models

Many authors have considered modeling hyponymy. However, in many cases, this task was

not the main point of their work, and we feel that the evaluation of the task has often been partially lacking. Here, we review several of those and attempt to shed a new light on existing results, based on the properties presented in section 2.

Several authors (Athiwaratkun and Wilson, 2018; Vendrov et al., 2015; Vilnis et al., 2018) have proposed Feature-vector embeddings of WORDNET. Among them, several have tested their embedding on the following task: they feed their model with the transitive closure of HYPONYMY, but withhold 4000 edges. They then test how many of those edges can be recovered by their model. They also test how many of 4000 random negative edges are correctly classified. They report the average of those numbers. We reproduce here their results for this task in Table 4a. As we see it, there are two issues with this task. First, it mainly accounts for recall, mostly ignoring precision. As we have explained in section 4, this can be a significant problem for WORDNET, which is sparse. Second, because WORDNET is the only input, it is questionable if any edge should be withheld at all (beyond those in the transitive closure of generating edges). We believe that, in this case, the gold standard to achieve is precisely the transitive closure. Indeed, because the graph presentation of WORDNET is nearly a tree, most of the time, the effect of removing an edge will be to detach a subtree. But, without any other source of information, this subtree could in principle be reattached to any node and still be a reasonable ontology, from a purely formal perspective. Thus we did not withhold any edge when training our second model on this task (the first one uses no edge at all). In turn, the numbers reported in Table 4a should not be taken too strictly.

Authors	Result
(Vendrov et al., 2015)	90.6
(Athiwaratkun and Wilson, 2018)	92.3
(Vilnis et al., 2018)	92.3
us, *fast*Text with LR and $\rho = 0.5$	87.2
us, single interval (tree-model)	94.5
us, interval disjunctions, $n = 5$	99.6

(a) Authors, systems and respective results on the task of detection of HYPONYMY in WORDNET

n	recall
1	0.91766
2	0.96863
5	0.99288
10	0.99973

(b) Typical recalls for multi-dimensional interval model. (Precision is always 1.)

Figure 4: Tables

7 Future Work and Conclusion

We found that defining the problem of representing HYPONYMY in a feature vector is not easy. Difficulties include 1. the sparseness of data, 2. whether one wants to base inclusion on an underlying (possibly relaxed) inclusion in the space of vectors, and 3. determining what one should generalize.

Our investigation of WORDNET over *fast*Text demonstrates that WORDNET classes are not cleanly linearly separated in *fast*Text, but they are sufficiently well separated to give a useful recall for an approximate inclusion property. Despite this, and because the negative cases vastly outnumber the positive cases, the rate of false negatives is still too high to give any reasonable precision. One could try to use more complex models, but the sparsity of the data would make such models extremely sensitive to overfitting.

Our second model takes a wholly different approach: we construct intervals directly from the HYPONYMY relation. The main advantage of this method is its simplicity and high-accuracy. Even with a single dimension it rivals other models. A possible disadvantage is that the multi-dimensional version of this model requires disjunctions to be performed. Such operations are not necessarily available in models which need to make use of the HYPONYMY relation. At this stage, we make no attempt to match the size of intervals to the probability of a word. We aim to address this issue in future work.

Finally, one could see our study as a criticism for using WORDNET as a natural representative of HYPONYMY: because WORDNET is almost structured like a tree, one can suspect that it in fact misses many hyponymy relations. This would also explain why our simple *fast*Text-based model predicts more relations than present in WORDNET. One could think of using other resources, such as JEUXDEMOTS (Lafourcade and Joubert, 2008). Yet our preliminary investigations suggest that these suffer from similar flaws — we leave a complete analysis to further work.

References

[Athiwaratkun and Wilson2018] Ben Athiwaratkun and Andrew Gordon Wilson. 2018. On modeling hierarchical data via probabilistic order embeddings. In *International Conference on Learning Representations*.

[Fellbaum1998] Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

[Lafourcade and Joubert2008] Mathieu Lafourcade and Alain Joubert. 2008. JeuxDeMots : un prototype ludique pour l'émergence de relations entre termes. In *JADT'08 : Journées internationales d'Analyse statistiques des Données Textuelles*, pages 657–666, France.

[Mikolov et al.2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119.

[Mikolov et al.2018] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[Nickel and Kiela2017] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6338–6347. Curran Associates, Inc.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

[Vendrov et al.2015] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2015. Order-embeddings of images and language. *CoRR*, abs/1511.06361.

[Vilnis and McCallum2014] Luke Vilnis and Andrew McCallum. 2014. Word representations via gaussian embedding. *CoRR*, abs/1412.6623.

[Vilnis et al.2018] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 263–272. Association for Computational Linguistics.