Evaluating Neural Network Semantics: Detecting Taxonomy with Embeddings

Klaudia Biczysko
Uppsala University, Sweden
Department of Linguistics and Philology
klaudiateresa.biczysko.4334@student.uu.se

Abstract

In this paper, we present our approach for the SemEval 2022 task 3 called *Presupposed Taxonomies: Evaluating Neural Network Semantics (PreTENS)*, more precisely a binary classification sub-task 1. The sub-task aims to predict the acceptability label for a sentence. Models need to not only identify if two nominal arguments hold the taxonomic relation (which can be defined as semantical hierarchy), but also pay attention to the context. We present an evaluation of supervised classifiers with additional features such as TF-IDF, Word2Vec ELMo and BERT. Thanks to our experiments, we managed to improve the accuracy scored by the baseline by introducing models such as Multi-layer Perceptron+BERT and Multi-layer Perceptron+ELMo. We also tried to test if our models can handle the domain change by testing on an external dataset.

1 Introduction

Achieving human-level language understanding in machines is one of the main objectives in Natural Language Understanding (NLU) (Linzen, 2020). A true understanding of a sentence's meaning i.e. understanding the structure and semantics, as well as the capacity to place it in the world is still a big challenge within the field of Natural Language Processing (Linzen, 2020).

An active subfield of NLU is investigating whether modern language models can recognize linguistic structures that are divergent at the semantic level. Researchers probe language models and their capacity to detect semantic relationships between words.

According to Caraballo (1999), the taxonomic (hierarchical) relation is one of the most important semantic relations, since it corresponds to the human cognitive paradigm of categorization. Taxonomies carry structured knowledge (Luu et al., 2016), providing valuable input for tasks, which are semantically intensive, such as question answering (Harabagiu and Hickl, 2006), textual entailment (Geffet and Dagan, 2005) and document clustering (Fodeh et al., 2011).

Using word embeddings as a semantic representation of words has been a major breakthrough for many Natural Language Processing tasks. Embeddings often lead to a better performance of language models and are also used for text classification tasks, as well for detecting semantic and linguistic relations between words (Tan et al., 2015).

This project is inspired by SemEval 2022 task 3 called *Presupposed Taxonomies: Evaluating Neural Network Semantics (PreTENS)*[1], more precisely a binary classification sub-task, which focuses on predicting the acceptability label for a sentence.

Our research has two aims. Firstly, we want to investigate whether neural networks (NNs) can recognize the taxonomic relation between two nominal arguments located in various statements, hence checking the potential of NNs as cognitive/linguistic models. The model needs to not only recognize the taxonomic relation between two nouns but also to verify if they hold that relationship in a given sentence since a dataset contains a wide range of presuppositional constructions. Secondly, the paper also aims to evaluate and compare context-free and contextualised embeddings in terms of carrying semantic relations. We believe that the embeddings of the nominal arguments are relevant since the arguments are the keywords in a sentence and if they hold the taxonomic relation, they should be close in a vector space.

In this paper, we demonstrate our methodology based on implementing TF-IDF (term frequency-inverse document frequency; proposed by the authors of the task) and both pre-trained word em-

[1] https://sites.google.com/view/semeval2022-pretens/

bedding techniques: traditional word embedding (Word2Vec (Mikolov et al., 2013a)) and contextual embedding (ELMo (Peters et al., 2018), BERT (Devlin et al., 2018)) as features for our classifiers: LinearSVC and Multi-layer Perceptron.

2 Related work

In order to identify semantic relations between words or concepts, many approaches have been researched. Hand-built semantic lexicons with the relation extracted from a dictionary or a corpus, such as WordNet (Miller et al., 1990), despite being created more than 20 years ago, are still widely used in modern NLP applications. Previous works on identifying taxonomic (hierarchical) relations between words or concepts are mostly based on datasets consisting of 'is-a' sentences describing hypernym-hyponym relation.

As Luu et al. (2016) points out the statistical, graph-based methods and linguistic approaches have been more researched than neural networks. By the linguistic approaches, we understand looking for lexical-syntactic patterns, while the statistical ones are based on the frequency of words that appear together. However, the accuracy level achieved by both the statistical and linguistic approaches is still not good enough. One of the problems behind the linguistic approaches is poor coverage since a wide variety of complex linguistic structures cannot be covered by the patterns (Luu et al., 2016). On the contrary, the statistical approaches have to some extent high coverage, however most of the time they are significantly reliant on the selection of feature types (Luu et al., 2016).

Recently, word embeddings have become a really important tool in Natural Language Processing research. They have been also used in detecting semantic and linguistic relations between words. Research has focused on learning the representations from word co-occurrence (similar embeddings indicate similar words), as presented by e.g. Tan et al. (2015), where researchers use word embeddings from the Word2Vec model (Mikolov et al., 2013a). Nonetheless, it was proven that learning co-occurrence based only on similarity is not sufficient for detecting taxonomies (Luu et al., 2016).

Yu et al. (2015) presented a supervised approach for learning word embeddings from pre-extracted taxonomic relationship data. However as Luu et al. (2016) pointed out, this technique relies on a training dataset detecting every taxonomic relation, which means that if a relation is not presented in the training set, then it will not be detected in the learning process. What's more, since they use pre-extracted pairs of hypernym-hyponyms, they do not take into consideration the contextual information between these pairs. The context has been proven to be important for detecting taxonomic relations between words (Yu et al., 2015).

Finally, Luu et al. (2016) proposed a method to learn term embeddings based on a dynamic weighting neural network, in order to encode not only the information about taxonomic relation (hypernym-hyponym), but also the contextual information between them. Consequently, the discovered embeddings are used as features in the supervised SVM to find the taxonomic relations.

3 Presupposed taxonomies

In this section, we describe linguistic definitions, which are crucial for a better understanding of the paper.

3.1 Taxonomy

For a better understanding of the term *taxonomy*, we may quote Ungerer and Schmid (2013), who explain that:

> Taxonomy is a hierarchical structure of units in terms of class inclusion such that superordinate units in the hierarchy include, or subsume, all items in subordinate units. In cognitive linguistics, taxonomies are hierarchies of concepts.

In other words, it is a science of classification. In the case of our project, two taxonomic relations are taken into consideration: *hyponomy* and *hyperonymy* (also known as *hypernymy*), where a hyponym denotes a subtype (subcategory), while a hypernym a supertype (supercategory). To exemplify it, in the sentence *dog is an animal*, *animal* can be identified as a supercategory of *dog*.

3.2 Presupposed taxonomies

When a natural language speaker designates linguistically information as being taken for granted, instead of being part of the main propositional content of a speech act, we talk about a phenomenon called presupposition (Beaver et al., 2021).

As the authors of the task underline, natural language consists of a great amount of two argument constructions, which restrict the taxonomic relationships and their order.

"Presupposition triggers" are expressions and formulations that contain presuppositions. Presuppositions may arise by default from the construction of a sentence or from specific words (Beaver et al., 2021). In the case of the dataset provided by the authors of the task, the construction of sentences may be marked as the presupposition trigger. For instance, the sentence *I like dogs, an interesting type of giraffe.* contradicts the presumptions that the two nominal arguments hold no taxonomic relationship (since *dogs* are not a type of *giraffe*).

4 Methodology

In this section, we first discuss the datasets, which were used in the experiments (section 4.1). We then briefly present the supervised classifiers (section 4.2), which we used to perform the binary classification on the mentioned datasets and four approaches for the features (section 4.3).

4.1 Datasets

There were two datasets used in the experiments. The first one was provided by the authors of the task, while the second one was created by us and built on the BLESS (Baroni and Lenci, 2011) dataset.

4.1.1 Original dataset

The authors of the task published datasets for three languages: English, Italian and French. However, in this research, only the English dataset was used. The dataset, provided by the authors of the task, consists of 3 folds, where each of the folds has between 1945 - 1948 artificially generated sentences. Sentences are constructed as generalizations (*I like A, and B in general*), comparatives (*I like A more than B*), exemplifications (*I like A, and in particular B*), but also as adversarial constructions (*I use A, but not B*). The idea behind the constructed sentences is that they impose presuppositions on the taxonomic status of the nominal arguments *A* and *B*. The nominal arguments are nouns from 30 semantic categories e.g. *trees, cars, animals*. The sentences are labelled with an acceptability label, where *1* means that a sentence is acceptable, while *0* means unacceptable.

In total, we have 5837 sentences including 3029 positive and 2808 negative sentences.

4.1.2 Domain-change test set

To check the models' behaviour in case of the domain change of nominal arguments in the test set, we decided to incorporate an external dataset published by Baroni and Lenci (2011). The BLESS (Baroni-Lenci Evaluation of Semantic Similarity) dataset consists of concrete nouns from different semantic categories e.g. *machines, animals, plants*. Each noun is linked to a set of words and grouped by the semantic relations, such as *hyperonymy*, *cohyponymy*, *meronymy*, *typical attribute*, *typical related event*, *random*, with each of them. We extracted the pairs of nouns with the relation marked as *hyperonymy*, *cohyponymy (coordinate terms)* or *random*. Next, we extracted the sentences from the task's dataset as our templates. We removed the nominal arguments and replaced them with the nouns from BLESS. The sentences were labelled with the same binary acceptability labels. In the end, the entire dataset was randomly shuffled. The finished test set consists of 1168 sentences since we decided to use hold-out cross-validation, where 20% of the entire dataset is a test set.

4.2 Classifiers

There were two supervised classifiers used in the experiments. The first one - a Linear Support Vector classifier - was proposed by the authors of the task in the baseline. Even though the research focuses on evaluating word embeddings, not classifiers, we wanted to test a simple neural network classifier. Multi-layer Perceptron was incorporated in the experiments due to its simplicity in comparison to other neural network models. Each classifier is implemented in the sklearn library, the machine learning library for Python.

LinearSVC The Linear Support Vector classifier is a support vector machine that generates a linear classifier. This algorithm conducts supervised classification using a linear kernel function. A typical approach to deal with the input is to use the character n-grams. LinearSVC using n-grams (up to three) as input features is used for the baseline. We decided to examine its abilities with word embeddings as input features. The specified hyperparameters of the classifier are presented in the table located in the appendix (see table 3).

Multi-layer Perceptron Perceptron is a supervised algorithm, which is used for binary classification tasks. The model is based on a biological neuron since it is built to resemble biological neurons' calculations. Since Perceptron could not be used for non-linear data due to having only one neuron (Minsky and Papert, 1969), Multi-layer Perceptron

(MLP) was constructed to address this limitation. In comparison to Perceptron, MLP may have one or more hidden layers. It is an example of a feedforward network, where the information travels from the input nodes through the hidden layer(-s) to the output nodes. Nowadays, Multi-layer Perceptrons are more practical (than Single-layer Perceptrons) for applications, hence we decided on using them in our experiments.

The Multi-layer Perceptron used in the calculations does not have any additional hyperparameters besides the default ones.

4.3 Features

In this section, we present four approaches for the features. First knowledge-based preprocessing (TF-IDF; section 4.3.1) is shortly defined, then we describe our three word embedding approaches (BERT, Word2Vec and ELMo; section 4.3.2).

4.3.1 TF-IDF-based approach

The TF-IDF (Term Frequency — Inverse Document Frequency) statistic is used to calculate the importance of the words in texts. The TF-IDF value is calculated by multiplying the term frequency and inverse document frequency values. The value of the TF-IDF measure rises in direct proportion to the number of times a word appears in the text. The number of documents that include the word in the entire corpus balance the TF-IDF value (Aizawa, 2003). As it was mentioned before, the TF-IDF-weighted character n-gram model was proposed by the authors of the task for the features for the classifier.

4.3.2 Word embedding-based approach

Word embeddings are high-dimensional representations of words. In other words, it is a technique where each word is converted into a vector (a numerical representation of the word). Below we present three word embedding-based approaches: one static (non-contextual; Word2Vec) and two contextual (BERT, ELMo). The main difference between static and contextual embedding is that the contextual approach takes word order into account.

Word2Vec Word2Vec is a static embedding approach implemented by Mikolov et al. (2013a) and Mikolov et al. (2013b). It is a word-based model, which takes words as input and outputs word embeddings. There are two architectures within Word2Vec: CBOW (Continuous Bag of Words) and Continous Skip-Gram Model. While both of them are predictive, CBOW predicts a target word from a list of context vectors and the Skip-Gram model does the opposite - it tries to predict the context words around the word which was taken as an input. The Skip-Gram technique tends to yield more accurate results on huge datasets Mikolov et al. (2013b).

ELMo ELMo (Embeddings from Language Models) is a type of deep contextualized word embeddings, which use bidirectional LSTMs trained on a language modelling objective (Peters et al., 2018). ELMo representations are contextual, deep and character-based. In comparison to static word embedding methods such as Word2Vec, ELMo takes the context of a word into consideration (contextual), hence the same word may have different word vectors in different sentences. It uses character convolutions (character-based) and can, therefore, out-of-vocabulary tokens. Finally, the word representations are deep, since all layers of a deep pre-trained neural network are combined and used for them.

BERT BERT is an acronym for Bidirectional Encoder Representations from Transformers and was introduced by Devlin et al. (2018). As ELMo, it is a type of pre-trained deep contextualized word embeddings, however, BERT uses Transformer, which is an attention-based model with positional encodings to represent word positions. The Transformer model is considered to be bi-directional since the encoder reads simultaneously the full word sequence. Hence this feature enables the model to learn the context of a word based on the left and right of the word. While ELMo is a character-based model, BERT encodes input as sub-words and learns embeddings for sub-words.

4.4 Evaluation metrics

To assess the quality of the presented models, we used metrics proposed by the authors of the task, such as accuracy, precision, recall, F1-measure and macro F-measure.

Since the results obtained by the models are high, the differences between them are not easily spotted. Hence we decided on including an error reduction rate, which was calculated against the error rate of the baseline. The error rate is the fraction of incorrect predictions divided by the total number of predictions. The error rate can be also calculated as a difference of accuracy from 100%.

To calculate the error reduction rate (ERR), we subtract the error rate of a model from the error rate of the baseline, dividing it by the error rate of the baseline. To get the percentage, we multiply it by 100%.

4.5 Hypotheses

Based on the above descriptions of the features and classifiers, we made below conclusions about our models: (1) A model with contextualised embeddings can detect presupposed taxonomies better than one with static embeddings due to taking context into consideration; (2) The static embeddings can obtain better scores in comparison to TF-IDF since NNs are more advanced; (3) When tested on the domain-change test set, models may obtain worse scores.

5 Experiments

In this section, we present our experiments. First, we describe experiments with the dataset provided by the authors of the task. We then describe experiments on the test set, which was described in the section 4.1.2.

5.1 Performance for the original dataset

For the dataset provided by the authors of the task, we have conducted four experiments to evaluate the performance of the models. Results are obtained by cross-validation: one fold is selected as a test, whereas the rest is merged (in case of more than 2 folds) to make training data. The obtained results of our experiments can be found in table 1.

Experiment 1: TF-IDF The authors of the task used a Linear Support Vector regressor with up to three n-grams as features for the baseline, which achieved an accuracy of 87.3%. We decided to test TF-IDF as features for both classifiers (LinearSVC and Multi-layer Perceptron) without additional preprocessing. TF-IDF was implemented by using TfidfVectorizer with the following parameters: (1) n-gram range between 1 and 3-grams, (2) exclusion of n-grams that have a document frequency strictly higher than 0.95, (3) exclusion of n-grams that occur in less than 3 sentences and (4) creation of a vocabulary that only consider the top 1000 max features ordered by term frequency across the corpus.

Experiment 2: Word2Vec As for our second experiment, we wanted to test a context-free word embedding and therefore settled on using Google's pre-trained Word2Vec model for feature extraction[2].

To pre-train vectors, part of the Google News dataset (about 100 billion words) was used. The model contains 300-dimensional vectors for 3 million words and phrases. The phrases were obtained using a simple data-driven approach (a Skip-gram model) introduced by (Mikolov et al., 2013a). Once again we tested our two classifiers: LinearSVC and Multi-layer Perceptron.

Experiment 3: ELMo For our third experiment, we decided to examine a contextualised embedding method - ELMo. To access ELMo, we made use of the TensorFlow Hub repository. After importing the pre-trained ELMo model v2, we extracted vectors for our sentences and returned the average of ELMo features. Then we used our ELMo vectors as features for the classifiers.

Experiment 4: BERT In our last experiment, we preprocessed our data by using another contextual embedding method - BERT - as features for LinearSVC and Multi-layer Perceptron. We decided on implementing library *BertEmbedding*, which was used on account of its simplicity. For our model, we chose the large BERT model pre-trained on unlabeled data extracted from the BooksCorpus with 800M words and English Wikipedia with 2,500M words. After obtaining the word embeddings, we averaged the results and used them as features.

5.2 Performance for the domain-change test set

As we mentioned in section 4.1.2, we decided on incorporating an external test set to evaluate the models' behaviour in case of the domain change of nominal arguments in the sentences. Since we wanted to keep our presupposition triggers (construction of sentences), the structure (and contexts) were left. Even though by leaving the structure of sentences, the ability of generalization of our models cannot be examined, models may be surprised by seeing new nominal arguments, which do not appear in a training set.

To test the domain-change dataset, we had to merge all 3 folds of the provided dataset to use it as the training set. Our domain-change dataset

[2]The open-source model is available here: GoogleNews-vectors-negative300.bin.gz.

Model	ERR	Accuracy	Precision	Recall	F1	F1macro
LinearSVC+TFIDF (baseline)	N/A	87.3%	84.6%	92.3%	88.3%	87.2%
LinearSVC+Word2Vec	-33.1%	83.1%	83.5%	84.1%	83.8%	83.1%
LinearSVC+ELMo	17.3%	89.5%	88.9%	91.3%	91%	89.5%
LinearSVC+BERT	37%	92%	92.3%	92.2%	92.3%	92%
MLP+TFIDF	48.8%	93.5%	94.2%	93.1%	93.7%	93.5%
MLP+Word2Vec	-159.8%	67%	84.6%	91%	87.7%	86.7%
MLP+ELMo	72.4%	96.5%	97.4%	95.9%	96.6%	96.5%
MLP+BERT	72.4%	96.5%	96.7%	96.5%	96.6%	96.5%

Table 1: Scores for the original dataset. The best performing models are bolded.

Model	ERR	Accuracy	Precision	Recall	F1	F1macro
LinearSVC+TFIDF (baseline)	N/A	74.4%	66.1%	100%	79.6%	72.6%
LinearSVC+Word2Vec	-26.9%	67.5%	63%	84.4%	72.2%	66.5%
LinearSVC+ELMo	-39.5%	64.3%	65.5%	60.1%	62.7%	64.2%
LinearSVC+BERT	-12.1%	71.3%	67.2%	83.2%	74.4%	70.9%
MLP+TFIDF	1.2%	74.7%	66.4%	100%	79.8%	73%
MLP+Word2Vec	-28.5%	67.1%	62.1%	88%	72.8%	65.6%
MLP+ELMo	-2%	73.9%	67.4%	92.6%	78%	72.9%
MLP+BERT	3.1%	75.2%	70.6%	86.1%	77.6%	74.9%

Table 2: Scores for the domain-change dataset. The best performing model is bolded.

was used as the test set. We tested all 8 models, which were described before, in hold-out cross-validation settings. The obtained results are given in table 2. Like in the case of the original dataset, LinearSVC+TF-IDF is the baseline.

6 Results & Analysis

The results and analysis are divided into three parts as follows: first, we describe our results for the SemEval sub-task, then we discuss our results from the domain-change experiments. In the end, we shortly analyze the differences between classifiers.

The main purpose of this work was to evaluate and compare embedding models when it comes to carrying semantic relations. A summary of the results is provided in tables 1 and 2, where table 1 describes the results for the original dataset, while table 2 for the domain-change dataset.

It is worth mentioning that knowledge-based preprocessing (TF-IDF) obtains good results with each classifier. However, It is surprising to see that TF-IDF using up to 3 n-grams scored that well because we hear that neural networks are better in general. Since our dataset is rather small, it does not meet the problem of sparsity. Moreover, semantic similarities in language are not taken into consideration by TF-IDF vectors. The weight for *dog* has no relationship with the weight for *animal*. Hence we would expect it to have worse scores, but in this dataset nominal arguments are repeated many times in different sentences in each of the fold, which could have affected the model and improved the accuracy.

It is significant that classifiers with Word2Vec as features have the lowest accuracy and hence no error reduction rate. The results seem to indicate that static embedding techniques such as Word2Vec perform much worse than contextual embedding techniques, especially when we compare MLP+Word2Vec and MLP+BERT/ELMo. The difference in error reduction rate between mentioned models is 255.9 points. The context has been proven to be important for detecting taxonomic relations between words, especially, when the construction of a sentence is a presupposition trigger. By not taking the order of words into consideration, a model may face a loss of semantic (and syntactic) understanding of the sentence.

The best performing models for the original dataset are MLP+BERT and MLP+ELMo, both of them have an error reduction rate of 72.4% while scoring accuracy of 96.5%. However, their

scores differ in two metrics: precision and recall. MLP+BERT has a higher recall, while MLP+ELMo scored higher precision. Both of these language models are contextual, which - as it was discussed before - is important in detecting semantic relations such as presupposed taxonomies.

As expected, the models tested on the external test set scored worse accuracy (see table 2). It is evident from the results that testing on another test set is a much harder task in comparison to basic binary classification. Surprisingly, the accuracy of LinearSVC+ELMo has significantly dropped, while models with BERT and TF-IDF maintain high results. We observe from table 2 that MLP+BERT is once again our best performing model for this task. It seems very probable that BERT owes its performance to the attention mechanism. Since BERT is built upon ELMO and it is a state-of-the-art in many downstream tasks, therefore it is more advanced and its scores come as no surprise for us. We assume that one of the reasons behind lower scores may be the fact that in the original data set terms are repetitive. As we mentioned before models have already seen all of the nominal arguments in the training data, just in different settings. A model can be kind of surprised by seeing different nominal arguments, what results seem to confirm.

We tested two classifiers: LinearSVC and MLP. In total, Multi-layer Perceptron yields better results than LinearSVC. We theorize that the reason behind this behaviour is the fact that the MLP classifier can better handle multiple boundaries, which separate our two classes since MLP can model e.g. XOR functions. Our dataset is a double challenge: models need to not only recognize the taxonomic relation between two nominal arguments but also to verify if they hold that relationship in a given sentence due to a wide range of presuppositional constructions. Since this is only a theory, we believe it is a place for further experiments, such as incorporating bigger datasets. It is worth remembering that NNs are still black boxes and hence Support Vector Machines classifiers are a lot easier to tune and interpret.

7 Conclusions & Future work

The problem of detecting taxonomy relations, such as hyperonym-hyponym, was studied. Two different approaches were implemented: the first one using knowledge-based preprocessing (TF-IDF) and the second one using static or contextual word embeddings (BERT, Word2Vec, ELMo). We have explored eight supervised classifiers for the binary classification task and showed that two models MLP+ELMo and MLP+BERT produced an error reduction rate of 72.4% on the original test set. A surprising outcome of the comparative study was that using knowledge-based preprocessing (TF-IDF) produces better scores than static word embedding.

In the second part of our experiments, we examined if the models we built have the ability to generalize. In comparison to the original dataset, the obtained scores are lower, which was expected. The best performing model is once again MLP+BERT.

To summarize, we have demonstrated that using pre-trained contextual embeddings as the features for classifiers highly outperforms traditional word embedding models. It's worth noticing that the knowledge-approach TF-IDF exceeded the pre-trained Word2Vec model. In general, we believe that neural networks have the ability to recognize the taxonomic relation.

Future research direction may include implementing more advanced neural networks such as Convolutional neural network (CNN) or Bidirectional long short-term memory (BiLSTM) as classifiers. Moreover, a dataset consisting of sentences, which are not artificially created and hence do not have a pattern, could be used. Additionally, French and Italian datasets could be tested. When it comes to the domain-change task, there is still a big room for improvement.

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Marco Baroni and Alessandro Lenci. 2011. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10.

David I. Beaver, Bart Geurts, and Kristie Denlinger. 2021. Presupposition. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Spring 2021 edition. Metaphysics Research Lab, Stanford University.

Sharon A Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 120–126.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Samah Fodeh, Bill Punch, and Pang-Ning Tan. 2011. On ontology-driven document clustering using core semantic features. *Knowledge and information systems*, 28(2):395–421.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 905–912.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? *arXiv preprint arXiv:2005.00955*.

Anh Tuan Luu, Yi Tay, Siu Cheung Hui, and See Kiong Ng. 2016. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 403–413.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Marvin Minsky and Seymour Papert. 1969. An introduction to computational geometry. *Cambridge tiass., HIT*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Liling Tan, Rohit Gupta, and Josef van Genabith. 2015. Usaar-wlv: Hypernym generation with deep neural nets. In *Proceedings of the 9th international workshop on semantic evaluation (semeval 2015)*, pages 932–937.

Friedrich Ungerer and Hans-Jorg Schmid. 2013. *An introduction to cognitive linguistics*. Routledge.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning term embeddings for hypernymy identification. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

A Appendix

Parameter	Value
random state	42
C	1.0
class weight	balanced
tol	0.0001

Table 3: Hyperparameters for the LinearSVC classifier.