Learning Term Embeddings for Taxonomic Relation Identification Using Dynamic Weighting Neural Network

Luu Anh Tuan
Institute for Infocomm Research, Singapore
at.luu@i2r.a-star.edu.sg

Yi Tay
Nanyang Technological University
ytay2@e.ntu.edu.sg

Siu Cheung Hui
Nanyang Technological University
asschui@ntu.edu.sg

See Kiong Ng
Institute for Infocomm Research, Singapore
skng@i2r.a-star.edu.sg

Abstract

Taxonomic relation identification aims to recognize the *'is-a'* relation between two terms. Previous works on identifying taxonomic relations are mostly based on statistical and linguistic approaches, but the accuracy of these approaches is far from satisfactory. In this paper, we propose a novel supervised learning approach for identifying taxonomic relations using term embeddings. For this purpose, we first design a *dynamic weighting neural network* to learn term embeddings based on not only the hypernym and hyponym terms, but also the contextual information between them. We then apply such embeddings as features to identify taxonomic relations using a supervised method. The experimental results show that our proposed approach significantly outperforms other state-of-the-art methods by 9% to 13% in terms of accuracy for both general and specific domain datasets.

1 Introduction

Taxonomies which serve as the backbone of structured knowledge are useful for many NLP applications such as question answering (Harabagiu et al., 2003) and document clustering (Fodeh et al., 2011). However, the hand-crafted, well-structured taxonomies including WordNet (Miller, 1995), OpenCyc (Matuszek et al., 2006) and Freebase (Bollacker et al., 2008) that are publicly available may not be complete for new or specialized domains. It is also time-consuming and error prone to identify taxonomic relations manually. As such, methods for automatic identification of taxonomic relations is highly desirable.

The previous methods for identifying taxonomic relations can be generally classified into two categories: statistical and linguistic approaches. The statistical approaches rely on the idea that frequently co-occurring terms are likely to have taxonomic relationships. While such approaches can result in taxonomies with relatively high coverage, they are usually heavily dependent on the choice of feature types, and suffer from low accuracy. The linguistic approaches which are based on lexical-syntactic patterns (e.g. *'A such as B'*) are simple and efficient. However, they usually suffer from low precision and coverage because the identified patterns are unable to cover the wide range of complex linguistic structures, and the ambiguity of natural language compounded by data sparsity makes these approaches less robust.

Word embedding (Bengio et al., 2001), also known as distributed word representation, which represents words with high-dimensional and real-valued vectors, has been shown to be effective in exploring both linguistic and semantic relations between words. In recent years, word embedding has been used quite extensively in NLP research, ranging from syntactic parsing (Socher et al., 2013a), machine translation (Zou et al., 2013) to sentiment analysis (Socher et al., 2013b). The current methods for learning word embeddings have focused on learning the representations from word co-occurrence so that similar words will have similar embeddings. However, using the co-occurrence based similarity learning alone is not effective for the purpose of identifying taxonomic relations.

Recently, Yu et al. (2015) proposed a super-

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 403–413,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

vised method to learn term embeddings based on pre-extracted taxonomic relation data. However, this method is heavily dependent on the training data to discover all taxonomic relations, i.e. if a pair of terms is not in the training set, it may become a negative example in the learning process, and will be classified as a non-taxonomic relation. The dependency on training data is a huge drawback of the method as no source can guarantee that it can cover all possible taxonomic relations for learning. Moreover, the recent studies (Velardi et al., 2013; Levy et al., 2014; Tuan et al., 2015) showed that contextual information between hypernym and hyponym is an important indicator to detect taxonomic relations. However, the term embedding learning method proposed in (Yu et al., 2015) only learns through the pairwise relations of terms without considering the contextual information between them. Therefore, the resultant quality is not good in some specific domain areas.

In this paper, we propose a novel approach to learn term embeddings based on *dynamic weighting neural network* to encode not only the information of hypernym and hyponym, but also the contextual information between them for the purpose of taxonomic relation identification. We then apply the identified embeddings as features to find the positive taxonomic relations using the supervised method SVM. The experimental results show that our proposed term embedding learning approach outperforms other state-of-the-art embedding learning methods for identifying taxonomic relations with much higher accuracy for both general and specific domains. In addition, another advantage of our proposed approach is that it is able to generalize from the training dataset the taxonomic relation properties for unseen pairs. Thus, it can recognize some true taxonomic relations which are not even defined in dictionary and training data. For the rest of this paper, we will discuss the proposed term embedding learning approach and its performance results.

2 Related work

Previous works on taxonomic relation identification can be roughly divided into two main approaches of statistical learning and linguistic pattern matching. Statistical learning methods include co-occurrence analysis (Lawrie and Croft, 2003), hierarchical latent Dirichlet allocation (LDA) (Blei et al., 2004; Petinot et al., 2011), clustering (Li et al., 2013), linguistic feature-based semantic distance learning (Yu et al., 2011), distributional representation (Roller et al., 2014; Weeds et al., 2014; Kruszewski et al., 2015) and co-occurrence subnetwork mining (Wang et al., 2013). Supervised statistical methods (Petinot et al., 2011) rely on hierarchical labels to learn the corresponding terms for each label. These methods require labeled training data which is costly and not always available in practice. Unsupervised statistical methods (Pons-Porrata et al., 2007; Li et al., 2013; Wang et al., 2013) are based on the idea that terms that frequently co-occur may have taxonomic relationships. However, these methods generally achieve low accuracies.

Linguistic approaches rely on lexical-syntactic patterns (Hearst, 1992) (e.g. *'A such as B'*) to capture textual expressions of taxonomic relations, and match them with the given documents or Web information to identify the relations between a term and its hypernyms (Kozareva and Hovy, 2010; Navigli et al., 2011; Wentao et al., 2012). These patterns can be manually created (Kozareva and Hovy, 2010; Wentao et al., 2012) or automatically identified (Snow et al., 2004; Navigli et al., 2011). Such liguistic pattern matching methods can generally achieve higher precision than the statistical methods, but they suffer from lower coverage. To balance the precision and recall, Zhu *et al.* (2013) and Tuan *et al.* (2014) have combined both unsupervised statistical and linguistic methods for finding taxonomic relations.

In recent years, there are a few studies on taxonomic relation identification using word embeddings such as the work of Tan et al. (2015) and Fu et al. (2014). These studies are based on word embeddings from the Word2Vec model (Mikolov et al., 2013a), which is mainly optimized for the purpose of analogy detection using co-occurrence based similarity learning. As such, these studies suffer from poor performance on low accuracy for taxonomic relation identification.

The approach that is closest to our work is the one proposed by Yu et al. (2015), which also learns term embeddings for the purpose of taxonomic relation identification. In the approach, a distance-margin neural network is proposed to learn term embeddings based on the pre-extracted taxonomic relations from the Probase database (Wentao et al., 2012). However, the neural network is trained using only the information of the term pairs (i.e. hypernym and hyponym) without considering the contextual information between them, which has been shown to be an important indicator for identifying taxonomic relations from previous studies (Velardi et al., 2013; Levy et al., 2014; Tuan et al., 2014). Moreover, if a pair of terms is not contained in the training set, there is high possibility that it will become a negative example in the learning process, and will likely be recognized as a non-taxonomic relation. The key assumption behind the design of this approach is not always true as no available dataset can possibly contain all taxonomic relations.

3 Methodology

In this section, we first propose an approach for learning term embeddings based on hypernym, hyponym and the contextual information between them. We then discuss a supervised method for identifying taxonomic relations based on the term embeddings.

3.1 Learning term embeddings

As shown in Figure 1, there are three steps for learning term embeddings: (i) extracting taxonomic relations; (ii) extracting training triples; and (iii) training neural network. First, we extract from WordNet all taxonomic relations as training data. Then, we extract from Wikipedia all sentences which contain at least one pair of terms involved in a taxonomic relation in the training data, and from that we identify the triples of hypernym, hyponym and contextual words between them. Finally, using the extracted triples as input, we propose a *dynamic weighting neural network* to learn term embeddings based on the information of these triples.

3.1.1 Extracting taxonomic relations

This step aims to extract a set of taxonomic relations for training. For this purpose, we use WordNet hierarchies for extracting all (direct and indirect) taxonomic relations between noun terms in WordNet. However, based on our experience, the relations involving with top-level terms such as ‘object’, ‘entity’ or ‘whole’ are usually ambiguous and become noise for the learning purpose. Therefore, we exclude from the training set all relations which involve with those top-level terms. Note that we also exclude from training set all taxonomic relations that are happened in the datasets used for testing in Section 4.1. As a result, the total number of extracted taxonomic relations is 236,058.

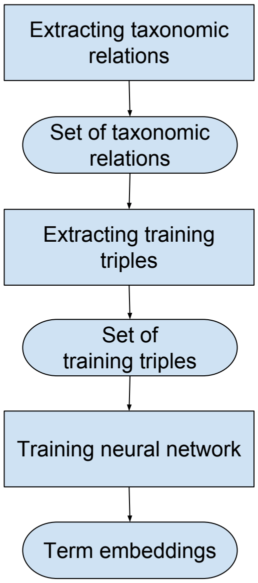
Extracting taxonomic relations
Set of taxonomic relations
Extracting training triples
Set of training triples
Training neural network
Term embeddings

Figure 1: Proposed approach for learning term embeddings.

3.1.2 Extracting training triples

This step aims to extract the triples of hypernym, hyponym and the contextual words between them. These triples will serve as the inputs to the neural network for training. In this research, we define contextual words as *all words located between the hypernym and hyponym in a sentence*. We use the latest English Wikipedia corpus as the source for extracting such triples.

Using the set of taxonomic relations extracted from the first step as reference, we extract from the Wikipedia corpus all sentences which contain at least two terms involved in a taxonomic relation. Specifically, for each sentence, we use the Stanford parser (Manning et al., 2014) to parse it, and check whether there is any pair of terms which are nouns or noun phrases in the sentence having a taxonomic relationship. If yes, we extract the hypernym, hyponym and all words between them from the sentence as a training triple. In total, we have extracted 15,499,173 training triples from Wikipedia.

Here, we apply the Stanford parser rather than matching the terms directly in the sentence in order to avoid term ambiguity as a term can serve for different grammatical functions such as noun or verb. For example, consider the following sentence:

- *Many supporters book tickets for the premiere of his new publication.*

The triple (*'publication', 'book', 'tickets for the premiere of his new'*) may be incorrectly added to the training set due to the occurrence of the taxonomic pair (*'publication'*, *'book'*), even though the meaning of *'book'* in this sentence is not about the *'publication'*.

3.1.3 Training neural network

Contextual information is an important indicator for detecting taxonomic relations. For example, in the following two sentences:

- *Dog is a type of animal which you can have as a pet.*
- *Animal such as dog is more sensitive to sound than human.*

The occurrence of contextual words *'is a type of'* and *'such as'* can be used to identify the taxonomic relation between *'dog'* and *'animal'* in the sentences. Many works in the literature (Kozareva and Hovy, 2010; Navigli et al., 2011; Wentao et al., 2012) attempted to manually find these contextual patterns, or automatically learn them. However, due to the wide range of complex linguistic structures, it is difficult to discover all possible contextual patterns between hypernyms and hyponyms in order to detect taxonomic relations effectively.

In this paper, instead of explicitly discovering the contextual patterns of taxonomic relations, we propose a *dynamic weighting neural network* to encode this information, together with the hypernym and hyponym, for learning term embeddings. Specifically, the target of the neural network is to predict the hypernym term from the given hyponym term and contextual words. The architecture of the proposed neural network is shown in Figure 2, which consists of three layers: input layer, hidden layer and output layer.

In our setting, the vocabulary size is V, and the hidden layer size is N. The nodes on adjacent layers are fully connected. Given a term/word t in the vocabulary, the input vector of t is encoded as a one-hot V-dimensional vector x_t, i.e. x_t consists of 0s in all elements except the element used to uniquely identify t which is set as 1. The weights between the input layer and output layer are represented by a $V \times N$ matrix W. Each row of W is a N-dimensional vector representation v_t of the associated word/term t of the input layer.

Given a hyponym term *hypo* and k context words c_1, c_2, .., c_k in the training triple, the output of hidden layer h is calculated as:

$$\begin{aligned} h &= W^{\top} \cdot \frac{1}{2k}(k \times x_{hypo} + x_{c_1} + x_{c_2} + ... + x_{c_k}) \\ &= \frac{1}{2k}(k \times v_{hypo} + v_{c_1} + v_{c_2} + ... + v_{c_k}) \end{aligned} \tag{1}$$

where v_t is the vector representation of the input word/term t.

The weight of h in Equation (1) is calculated as the average of the vector representation of hyponym term and contextual words. Therefore, this weight is not based on a fixed number of inputs. Instead, it is dynamically updated based on the number of contextual words k in the current training triple, and the hyponym term. This model is called *dynamic weighting neural network* to reflect its dynamic nature. Note that to calculate h, we also multiply the vector representation of hyponym by k to reduce the bias problem of high number of contextual words, so that the weight of the input vector of hyponym is balanced with the total weight of contextual words.

From the hidden layer to the output layer, there is another weight $N \times V$ for the output matrix W'. Each column of W' is a N-dimensional vector v'_t representing the output vector of t. Using these weights, we can compute an output score u_t for each term/word t in the vocabulary:

$$u_t = {v'_t}^{\top} \cdot h \tag{2}$$

where v'_t is the output vector of t.

We then use soft-max, a log-linear classification model, to obtain the posterior distribution of hypernym terms as follows:

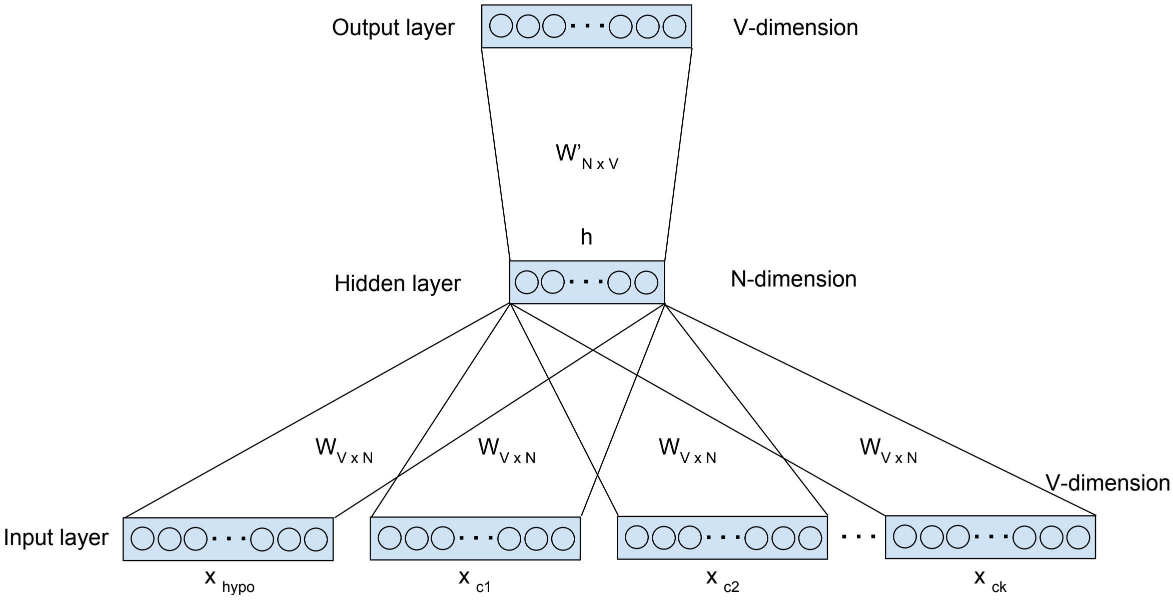

Figure 2: The architecture of the proposed dynamic weighting neural network model.

$$\begin{aligned} & p(hype|hypo, c_1, c_2, .., c_k) \\ & = \frac{e^{u_{hype}}}{\sum_{i=1}^{V} e^{u_i}} \\ & = \frac{e^{v'^{\top}_{hype} \cdot \frac{1}{2k}(k \times v_{hypo} + \sum_{j=1}^{k} v_{c_j})}}{\sum_{i=1}^{V} e^{v'^{\top}_{i} \cdot \frac{1}{2k}(k \times v_{hypo} + \sum_{j=1}^{k} v_{c_j})}} \end{aligned} \tag{3}$$

The objective function is then defined as:

$$O = \frac{1}{T} \sum_{t=1}^{T} log(p(hype_t|hypo_t, c_{1t}, c_{2t}, .., c_{kt})) \tag{4}$$

where T is the number of training triples; $hype_t$, $hypo_t$ and c_{it} are hypernym term, hyponym term and contextual words respectively in the training triple t.

After maximizing the log-likelihood objective function in Equation (4) over the entire training set using stochastic gradient descent, the term embeddings are learned accordingly.

3.2 Supervised taxonomic relation identification

To decide whether a term x is a hypernym of term y, we build a classifier that uses embedding vectors as features for taxonomic relation identification. Specifically, we use Support Vector Machine (SVM) (Cortes and Vapnik, 1995) for this purpose. Given an ordered pair (x, y), the input feature is the concatenation of embedding vectors (v_x,v_y) of x and y. In addition, our term embedding learning approach has the property that the embedding of hypernym is encoded based on not only the information of hyponym but also the information of contextual words. Therefore, we add one more feature to the input of SVM, i.e. the offset vector $(v_x - v_y)$, to contain the information of all contextual words between x and y. In summary, the feature vector is a $3d$ dimensional vector $\langle v_x, v_y, v_x - v_y \rangle$, where d is the dimension of term embeddings. As will be shown later in the experimental results, the offset vector plays an important role in the task of taxonomic relation identification of our approach.

4 Experiments

We conduct experiments to evaluate the performance of our term embedding learning approach on the general domain areas as well as the specific domain areas. In performance evaluation, we compare our approach with two other state-of-the-art supervised term embedding learning methods in Yu et al. (2015) and the Word2Vec model (Mikolov et al., 2013a).

4.1 Datasets

There are five datasets used in the experiments. Two datasets, namely BLESS and ENTAILMENT, are general domain datasets. The other three datasets, namely Animal, Plant and Vehicle, are specific domain datasets.

- BLESS (Baroni and Lenci, 2011) dataset: It covers 200 distinct, unambiguous concepts (terms); each of which is involved with other terms, called *relata*, in some relations. We extract from BLESS 14,547 pairs of terms for the following four types of relations: taxonomic relation, meronymy relation (a.k.a. part-of relation), coordinate relation (i.e. two terms having the same hypernym), and random relation. From these pairs, we set taxonomic relations as positive examples, while other relations form the negative examples.
- ENTAILMENT dataset (Baroni et al., 2012): It consists of 2,770 pairs of terms, with equal number of positive and negative examples of taxonomic relations. Altogether, there are 1,376 unique hyponyms and 1,016 unique hypernyms.
- Animal, Plant and Vehicle datasets (Velardi et al., 2013): They are taxonomies constructed based on the dictionaries and data crawled from the Web for the corresponding domains. The positive examples are created by extracting all possible (direct and indirect) taxonomic relations from the taxonomies. The negative examples are generated by randomly pairing two terms which are not involved in any taxonomic relation.

The number of terms, positive examples and negative examples extracted from the five datasets are summarized in Table 1.

Dataset	# terms	# positive	# negative
BLESS	5229	1337	13210
ENTAILMENT	2392	1385	1385
Animal	659	4164	8471
Plant	520	2266	4520
Vehicle	117	283	586

Table 1: Datasets used in the experiments.

4.2 Comparison models

In the experiments, we use the following supervised models for comparison:

- SVM+Our: This model uses SVM and the term embeddings obtained by our learning approach. The input is a 3d-dimensional vector $\langle v_x, v_y, v_x - v_y \rangle$, where d is the dimension of term embeddings, x and y are two terms used to check whether x is a hypernym of y or not, and v_x, v_y are the term embeddings of x and y respectively.
- SVM+Word2Vec: This model uses SVM and the term embeddings obtained by applying the Skip-gram model (Mikolov et al., 2013a) on the entire English Wikipedia corpus. The input is also a 3d-dimensional vector as in the SVM+Our model. Note that the results of the Skip-gram model are word embeddings. So if a term is a multiword term, its embedding is calculated as the average of all words in the term.
- SVM+Yu: This model uses SVM and the term embeddings obtained by using Yu et al.'s method (2015). According to the best setting stated in (Yu et al., 2015), the input is a 2d+1 dimensional vector $\langle O(x), E(y), \|O(x)\text{-}E(y)\|_1 \rangle$, where $O(x)$, $E(y)$ and $\|\text{O(x)-E(y)}\|_1$ are hyponym embedding of x, hypernym embedding of y and 1-norm distance of the vector ($O(x)$-$E(y)$) respectively.

Parameter settings. The SVM in the three models is trained using a RBF kernel with λ= 0.03125 and penalty term C = 8.0. For term embedding learning, the vector's dimension is set to 100. The tuning of the dimension will be discussed in Section 4.6.

4.3 Performance on general domain datasets

For the general domain datasets, we have conducted two experiments to evaluate the performance of our proposed approach.

Experiment 1. For the BLESS dataset, we hold out one concept for testing and train on the remaining 199 concepts. The hold-out concept and its relatum constitute the testing set, while the remaining 199 concepts and their relatum constitute the training set. To further separate the training and testing sets, we exclude from the training set any pair of terms that has one term appearing in the testing set. We report the average accuracy across all concepts. For the ENTAILMENT dataset, we use the same evaluation method: hold out one hypernym for testing and train on the remaining hypernyms, and we also report the average accuracy across all hypernyms. Furthermore, to evaluate the effect of the offset vector to taxonomic relation identification, we deploy a setting that removes the offset vector in the feature vectors of SVM. Specifically, for SVM+Our and SVM+Word2Vec, the input vector is changed from $\langle v_x, v_y, v_x - v_y\rangle$ to $\langle v_x, v_y\rangle$. We use the subscript $short$ to denote this setting.

Model	Dataset	Accuracy
SVM+Yu	BLESS	90.4%
SVM+Word2Vec$_{short}$	BLESS	83.8%
SVM+Word2Vec	BLESS	84.0%
SVM+Our$_{short}$	BLESS	91.1%
SVM+Our	BLESS	**93.6%**
SVM+Yu	ENTAIL	87.5%
SVM+Word2Vec$_{short}$	ENTAIL	82.8%
SVM+Word2Vec	ENTAIL	83.3%
SVM+Our$_{short}$	ENTAIL	88.2%
SVM+Our	ENTAIL	**91.7%**

Table 2: Performance results for the BLESS and ENTAILMENT datasets.

Table 2 shows the performance of the three supervised models in Experiment 1. Our approach achieves significantly better performance than Yu's method and Word2Vec method in terms of accuracy (t-test, p-value < 0.05) for both BLESS and ENTAILMENT datasets. Specifically, our approach improves the average accuracy by 4% compared to Yu's method, and by 9% compared to the Word2Vec method. The Word2Vec embeddings have the worst result because it is based only on co-occurrence based similarity, which is not effective for the classifier to accurately recognize all the taxonomic relations. Our approach performs better than Yu's method and it shows that our approach can learn embeddings more effectively. Our approach encodes not only hypernym and hyponym terms but also the contextual information between them, while Yu's method ignores the contextual information for taxonomic relation identification.

Moreover, from the experimental results of SVM+Our and SVM+Our$_{short}$, we can observe that the offset vector between hypernym and hyponym, which captures the contextual information, plays an important role in our approach as it helps to improve the performance in both datasets. However, the offset feature is not so important for the Word2Vec model. The reason is that the Word2Vec model is targeted for the analogy task rather than taxonomic relation identification.

Experiment 2. This experiment aims to evaluate the generalization capability of our extracted term embeddings. In the experiment, we train the classifier on the BLESS dataset, test it on the ENTAILMENT dataset and vice versa. Similarly, we exclude from the training set any pair of terms that has one term appearing in the testing set. The experimental results in Table 3 show that our term embedding learning approach performs better than other methods in accuracy. It also shows that the taxonomic properties identified by our term embedding learning approach have great generalization capability (i.e. less dependent on the training set), and can be used generically for representing taxonomic relations.

Model	Training	Testing	Accuracy
SVM+Yu	BLESS	ENTAIL	83.7%
SVM+Word2Vec$_{short}$	BLESS	ENTAIL	76.5%
SVM+Word2Vec	BLESS	ENTAIL	77.1%
SVM+Our$_{short}$	BLESS	ENTAIL	85.8%
SVM+Our	BLESS	ENTAIL	**89.4%**
SVM+Yu	ENTAIL	BLESS	87.1%
SVM+Word2Vec$_{short}$	ENTAIL	BLESS	78.0%
SVM+Word2Vec	ENTAIL	BLESS	78.9%
SVM+Our$_{short}$	ENTAIL	BLESS	87.1%
SVM+Our	ENTAIL	BLESS	**90.6%**

Table 3: Performance results for the general domain datasets when using one domain for training and another domain for testing.

4.4 Performance on specific domain datasets

Similarly, for the specific domain datasets, we have conducted two experiments to evaluate the performance of our proposed approach.

Experiment 3. For each of the Animal, Plant and Vehicle datasets, we also hold out one term for testing and train on the remaining terms. The positive and negative examples which contain the hold-out term constitute the testing set, while other positive and negative examples constitute the training set. We also exclude from the training set any pair of terms that has one term appearing in the testing set. The experimental results are given in Table 4. We can observe that not only for general domain datasets but also for specific domain datasets, our term embedding learning approach has achieved significantly better performance than Yu's method and the Word2Vec method in terms of accuracy (t-test, p-value < 0.05). Specifically, our approach improves the average accuracy by 22% compared to Yu's method, and by 9% compared to the Word2Vec method.

Model	Dataset	Accuracy
SVM+Yu	Animal	67.8%
SVM+Word2Vec	Animal	80.2%
SVM+Our	Animal	**89.3%**
SVM+Yu	Plant	65.7%
SVM+Word2Vec	Plant	81.5%
SVM+Our	Plant	**92.1%**
SVM+Yu	Vehicle	70.5%
SVM+Word2Vec	Vehicle	82.1%
SVM+Our	Vehicle	**89.6%**

Table 4: Performance results for the Animal, Plant and Vehicle datasets.

Another interesting point to observe is that the accuracy of Yu's method drops significantly in specific domain datasets (as shown in Table 4) when compared to the general domain datasets (as shown in Table 2). One possible explanation is the accuracy of Yu's method depends on the training data. As Yu's method learns the embeddings using pre-extracted taxonomic relations from Probase, and if a relation does not exist in Probase, there is high possibility that it becomes a negative example and be recognized as a non-taxonomic relation by the classifier. Therefore, the training data extracted from Probase plays an important role in Yu's method. For general domain datasets (BLESS and ENTAILMENT), there are about 75%-85% of taxonomic relations in these datasets found in Probase, while there are only about 25%-45% of relations in the specific domains (i.e. Animal, Plant and Vehicle) found in Probase. Therefore, Yu's method achieves better performance in general domain datasets than the specific ones. Our approach, in contrast, less depends on the training relations. Therefore, it can achieve high accuracy in both the general and specific domain datasets.

Experiment 4. Similar to experiment 2, this experiment aims to evaluate the generalization capability of our term embeddings. In this experiment, for each of the Animal, Plant and Vehicle domains, we train the classifier using the positive and negative examples in each domain and test the classifier in other domains. The experimental results in Table 5 show that our approach achieves the best performance compared to other state-of-the-art methods for all the datasets. As also shown in Table 3, our approach has achieved high accuracy for both general and specific domain datasets, while in Yu's method, there is a huge difference in accuracy between these domain datasets.

Model	Training	Testing	Accuracy
SVM+Yu	Animal	Plant	65.5%
SVM+Word2Vec	Animal	Plant	82.4%
SVM+Our	Animal	Plant	**91.9%**
SVM+Yu	Animal	Vehicle	66.2%
SVM+Word2Vec	Animal	Vehicle	81.3%
SVM+Our	Animal	Vehicle	**89.5%**
SVM+Yu	Plant	Animal	68.4%
SVM+Word2Vec	Plant	Animal	81.8%
SVM+Our	Plant	Animal	**91.5%**
SVM+Yu	Plant	Vehicle	65.2%
SVM+Word2Vec	Plant	Vehicle	81.0%
SVM+Our	Plant	Vehicle	**88.5%**
SVM+Yu	Vehicle	Animal	70.9%
SVM+Word2Vec	Vehicle	Animal	79.7%
SVM+Our	Vehicle	Animal	**87.6%**
SVM+Yu	Vehicle	Plant	66.2%
SVM+Word2Vec	Vehicle	Plant	78.7%
SVM+Our	Vehicle	Plant	**87.7%**

Table 5: Performance results for the specific domain datasets when using one domain for training and another domain for testing.

4.5 Empirical comparison with WordNet

By error analysis, we found that our results may complement WordNet. For example, in the Animal domain, our approach identifies *'wild sheep'* as a hyponym of *'sheep'*, but in WordNet, they are siblings. However, many references [1], [2] consider *'wild sheep'* as a species of *'sheep'*. Another such example is shown in the Plant domain, where our ap-

[1] http://en.wikipedia.org/wiki/Ovis

[2] http://www.bjornefabrikken.no/side/norwegian-sheep/

proach recognizes *'lily'* as a hyponym of *'flowering plant'*, but WordNet places them in different subtrees incorrectly [3]. Therefore, our results may help restructure and even extend WordNet.

Note that these taxonomic relations are not in our training set. They are also not recognized by the term embeddings obtained from the Word2Vec method and Yu et al.'s method. It again shows that our term embedding learning approach has the capability to identify taxonomic relations which are not even defined in dictionary or training data.

4.6 Tuning vector dimensions

We also conduct experiments to learn term embeddings from the general domain datasets with different dimensions (i.e. 50, 100, 150 and 300) using our proposed approach. We then use these embeddings to evaluate the performance of taxonomic relation identification based on training time and accuracy, and show the results in Table 6. The experiments are carried out on a PC with Intel(R) Xeon(R) CPU at 3.7GHz and 16GB RAM.

Dimension	Dataset	Training time	Accuracy
50	BLESS	1825s	87.7%
100	BLESS	2991s	89.4%
150	BLESS	4025s	89.9%
300	BLESS	7113s	90.0%
50	ENTAIL	1825s	88.5%
100	ENTAIL	2991s	90.6%
150	ENTAIL	4025s	90.9%
300	ENTAIL	7113s	90.9%

Table 6: Performance results based on training time and accuracy of the SVM+Our model using different vector dimensions.

In general, when increasing the vector dimension, the accuracy of our term embedding learning approach will be increased gradually. More specifically, the accuracy improves slightly when the dimension is increased from 50 to 150. But after that, increasing the dimension has very little effect on the accuracy. We observe that the vector dimension for learning term embeddings can be set between 100 to 150 to achieve the best performance, based on the trade-off between accuracy and training time.

5 Conclusion

In this paper, we proposed a novel approach to learn term embeddings using dynamic weighting neural network. This model encodes not only the hypernym and hyponym terms, but also the contextual information between them. Therefore, the extracted term embeddings have good generalization capability to identify unseen taxonomic relations which are not even defined in dictionary and training data. The experimental results show that our approach significantly outperforms other state-of-the-art methods in terms of accuracy in identifying taxonomic relation identification.

References

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

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32.

Yoshua Bengio, Rjean Ducharme, and Pascal Vincent. 2001. A Neural Probabilistic Language Model. *Proceedings of the NIPS conference*, pages 932–938.

David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. 2004. Hierarchical topic models and the nested chinese restaurant process. *Advances in Neural Information Processing Systems*, pages 17–24.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

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

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. *Proceedings of the 52nd Annual Meeting of the ACL*, pages 1199–1209.

Sanda M. Harabagiu, Steven J. Maiorano, and Marius A. Pasca. 2003. Open-domain textual question an-

[3] https://en.wikipedia.org/wiki/Lilium

swering techniques. *Natural Language Engineering*, 9(3):231–267.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545.

Zornitsa Kozareva and Eduard Hovy. 2010. A Semi-supervised Method to Learn and Construct Taxonomies Using the Web. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1110–1118.

German Kruszewski, Denis Paperno, and Marco Baroni. 2015. Deriving boolean structures from distributional vectors. *Transactions of the Association for Computational Linguistics*, 3:375–388.

Dawn J. Lawrie and W. Bruce Croft. 2003. Generating hierarchical summaries for web searches. *Proceedings of the 26th ACM SIGIR conference*, pages 457–463.

Omer Levy, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. 2014. Do supervised distributional methods really learn lexical inference relations. *Proceedings of the NAACL conference*, pages 1390–1397.

Baichuan Li, Jing Liu, Chin Y. Lin, Irwin King, and Michael R. Lyu. 2013. A Hierarchical Entity-based Approach to Structuralize User Generated Content in Social Media: A Case of Yahoo! Answers. *Proceedings of the EMNLP conference*, pages 1521–1532.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. *Proceedings of the 52nd Annual Meeting of the ACL*, pages 55–60.

Cynthia Matuszek, John Cabral, Michael J. Witbrock, and John DeOliveira. 2006. An introduction to the syntax and content of cyc. *Proceedings of the AAAI Spring Symposium*, pages 44–49.

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

George A. Miller. 1995. WordNet: a Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A Graph-based Algorithm for Inducing Lexical Taxonomies from Scratch. *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1872–1877.

Yves Petinot, Kathleen McKeown, and Kapil Thadani. 2011. A hierarchical model of web summaries. *Proceedings of the 49th Annual Meeting of the ACL*, pages 670–675.

Aurora Pons-Porrata, Rafael Berlanga-Llavori, and Jose Ruiz-Shulcloper. 2007. Topic discovery based on text mining techniques. *Information processing & management*, 43(3):752–768.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. *Proceedings of the COLING conference*, pages 1025–1036.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013a. Parsing with compositional vector grammars. *Proceedings of the 51st Annual Meeting of the ACL*, pages 932–937.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the EMNLP conference*, pages 1631–1642.

Liling Tan, Rohit Gupta, and Josef van Genabith. 2015. Usaar-wlv: Hypernym generation with deep neural nets. *Proceedings of the SemEval*, pages 932–937.

Luu A. Tuan, Jung J. Kim, and See K. Ng. 2014. Taxonomy Construction using Syntactic Contextual Evidence. *Proceedings of the EMNLP conference*, pages 810–819.

Luu A. Tuan, Jung J. Kim, and See K. Ng. 2015. Incorporating Trustiness and Collective Synonym/Contrastive Evidence into Taxonomy Construction. *Proceedings of the EMNLP conference*, pages 1013–1022.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.

Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. 2013. A phrase mining framework for recursive construction of a topical hierarchy. *Proceedings of the 19th ACM SIGKDD conference*, pages 437–445.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David J Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. *Proceedings of the COLING conference*, pages 2249–2259.

Wu Wentao, Li Hongsong, Wang Haixun, and Kenny. Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. *Proceedings of the ACM SIGMOD conference*, pages 481–492.

Jianxing Yu, Zheng-Jun Zha, Meng Wang, Kai Wang, and Tat-Seng Chua. 2011. Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. *Proceedings of the EMNLP conference*, pages 140–150.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning term embeddings for hypernymy identification. *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1390–1397.

Xingwei Zhu, Zhao Y. Ming, and Tat-Seng Chua. 2013. Topic hierarchy construction for the organization of multi-source user generated contents. *Proceedings of the 36th ACM SIGIR conference*, pages 233–242.

Will Y Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. *Proceedings of the EMNLP conference*, pages 1393–1398.