Detecting General Word in a Word Pair

Sayan Mandal — Avinash Gupta — Shubham Kumar
Pate Vedant Sanjay —Aditya Bhagwat

November 2017

1 Introduction

Modeling hypernymy, such as dog is-an animal, is an important generalization aid to many NLP tasks, such as entailment, relation extraction, and question answering. Supervised learning from labeled hypernym sources, such as WordNet, limit the coverage of these models, which can be addressed by learning hypernyms from unlabeled text.Knowledge of the hyponymy relation is critical for tasks such as Question Answering , Natural Language Inference, and Coreference Resolution.Numerous applications benefit from compactly representing context distributions, which assign meaning to objects under the rubric of distributional semantics. In natural language processing, distributional semantics has long been used to assign meanings to words (that is, to lexemes in the dictionary, not individual instances of word tokens). The meaning of a word in the distributional sense is often taken to be the set of textual contexts (nearby tokens) in which that word appears, represented as a large sparse bag of words (SBOW).In this work, we have tried to exploit the difference in context properties of general word and specific word and tried to devise some unsupervised algorithm to detect Hypernym Hyponym relation in a word pair.

2 Prior Work

2.1 Chasing Hypernyms in Vector Spaces with Entropy

In this paper, Authors introduced SLQS, a new entropy-based measure for the unsupervised identification of hypernymy and its directionality in Distributional Semantic Models (DSMs).
One of the first proposed measures formalizing the DIH is WeedsPrec (Weeds and Weir, 2003; Weeds et al., 2004), which quantifies the weights of the features f of a narrow term u that are included into the set of features of a broad term v:

$$WeedsPrec(u,v) = \frac{\sum_{f \in F_u \cap F_v} w_u(f)}{\sum_{f \in F_u} w_u(f)}$$

where Fx is the set of features of a term x, and wx(f) is the weight of the feature f of the term x. Variations of this measure have been introduced by Clarke (2009), Kotlerman et al. (2010) and Lenci and Benotto (2012).

Their Hypothesis were:

- SLQS, which measures the semantic generality of a word by the entropy of its statistically most prominent contexts.
- Hypernyms are semantically more general than hyponyms.
- Hypernyms are more frequent than hyponyms.
- Hyponyms and hypernyms are distributionally similar, but hyponyms tend to occur in more informative contexts than hypernyms.

For every word wi they identified their top N (which was set to be 50) contexts and for every contexts they measured their entropy according to the following formula:

$$H(c) = -\sum_{i=1}^{n} p(f_i|c) \cdot log_2(p(f_i|c))$$

where $p(fi \mid c)$ is the probability of the feature fi given the context c . Then the resultant value was scaled down to 0-1 range using min-maxscaling.

For each term wi they found the Median entropy of its N contexts . And then following measurement to infer:

$$SLQS(w_1, w_2) = 1 - \frac{E_{w_1}}{E_{w_2}}$$

if SLQS(w1,w2)¿0, w1 is semantically less general than w2. SLQS scored a precision of 87%.

2.2 Learning Hypernymy Over Word Embeddings

It is a Supervised Method.Word embeddings such as GloVe and Word2Vec have shown promise in a variety of NLP tasks.These word representations are constructed to minimise the distance between words with similar contexts. According to the distribution Hypothesis , this means that words with similar meanings should have similar representations.It has been shown that continuous word embeddings encode various simple lexical relations, such as singular-plural or country-capital as offsets in vector space .This Group at Stanford has worked on the concept of Word Embeddings to detect the Hyponym-Hypernymy relation.

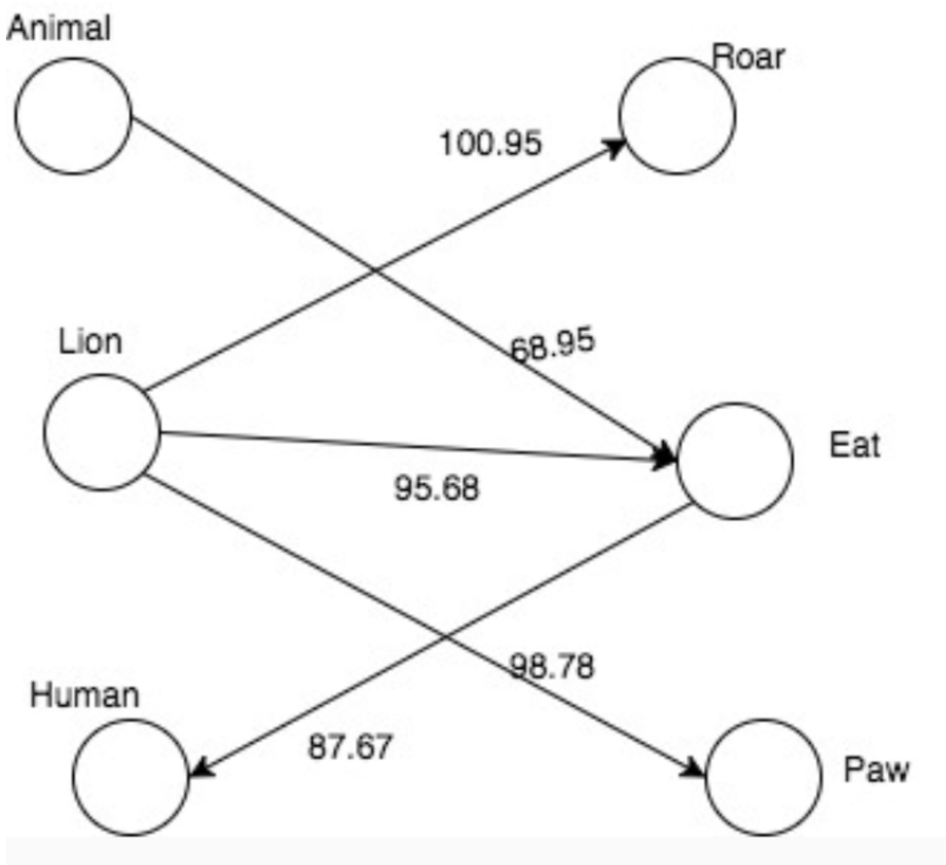
2.3 Classification task

A pair of words (w1, w2) is given, and the objective is to classify the pair as a positive example if w2 is a hypernym of w1, or a negative example otherwise.Classification of hypernyms has been carried out with some success using co-occurrence-based vectors.This Group has used the concept of Deep Learning to train their model.
They used BLESS and WORDNET as the Dataset.They used input as the subtraction of the word vectors of two words and used a Fully connected ANN with a hidden layer consisting of 500 neurons and a softmax output layer to train their model.Using GloVe embeddings they got an accuracy of 93.0% on the test set.

3 Dataset Description

We have derived four Files corresponding to the following:

- Word Nodes
- Context Nodes
- Word Context Edges
- Context Word edges



An example of the dataset model is given above.

4 Methodologies

Four Unsupervised Methodologies Followed by us are described follow:

4.1 First Measure : Average Degree of Context

Hypothesis: General word has more general context and hence each context word should be connected to more words.

Process :

- In this measure, we take each word in the given word pair and then find their contexts respectively.
- Then,we find the sum of degree of each context for the two words.
- Finally , we find the average degree of the contexts and compare them.

- The word with more degree should be the more general word.

4.2 Second Measure : Number of neighbours of context

Hypothesis: General word has more general context and hence more words should have at least one context word of the more general word in their context.

Process :

- In this measure, we take each word in the given word pair and then find their contexts.
- Then,we find the set of words which share at least one context word with the two words respectively..
- Then,we compare the number of distinct words in the two sets found in the last step.
- The word corresponding to the bigger set should be the more general word.

4.3 Third Measure : Context of specific word should be a subset of the general word

Hypothesis: Most of the words which are in the context of the specific word also be present in the context of the general word.

Process :

- In this measure, we take each word in the given word pair and then find their contexts respectively.
- Then, we find the set difference of the context for both the words and get two new sets(A-B and B-A).
- Finally , we compare the number of distinct words in the two sets we found in the last step.
- The word corresponding to the bigger set should be the more general word.

Problem: The number of words in A-B and B-A was coming out to be the same since our dataset has top 1000 contexts of a word.

Solution: We did Thresholding. We have taken the top n neighbours corresponding to a node sorted by the edge weight between them.

4.4 Fourth Measure : Degree Entropy Of the Network

Hypothesis: The network of the more general word should have more degree entropy as the range of degrees of the context words will be large compared to the specific word

Process:

- In this measure, we take each word in the given word pair and then find their contexts.
- Then, we find all the words connected to at least one word in the context and hence, we get two networks.
- Then, we find the degree distribution of nodes corresponding to the two networks.
- Finally , we find the degree entropy of the two networks.
- The word corresponding to the network with more degree entropy should be the more general word.

We also used the above measure considering only the nouns, verbs and adjectives respectively and hence got 3 more comparisons.

NOTE : For the remaining three comparisons we will only consider the degree entropy in the context of the words to compare. I.e. while making the degree distribution just consider the context nodes.

5 Results

5.1 TestSet:

We have extracted 1309 Hypernym-Hyponym pair as ground truth from the BLESS dataset where first word denotes the Hypernym and second word Hyponym e.g. Animal-Lion. We evaluated our methodlogies on this testset. Results are given for different approaches below.

Measure	Threshold	No. of pairs	Positive	Accuracy
First	1000	1309	877	67.00%
First	500	1309	809	61.80%
First	200	1309	671	51.26%
First	100	1309	620	47.36%
Second	1000	1309	670	51.18%
Third	1000	1309	539	45.30%
Third (Adj.)	1000	1309	947	72.34%
Third (Noun)	1000	1309	603	46.52%
Third (Verb)	1000	1309	759	58.18%

Measure	Threshold	No. of pairs	Positive	Accuracy
Fourth	1000	1309	967	73.87%
Fourth	700	1309	997	76.16%
Fourth	**500**	**1309**	**1008**	**77.00%**
Fourth	300	1309	995	76.00%
Fourth	200	1309	921	70.35%
Fourth	100	1309	865	66.08%
Fourth (Noun)	1000	1309	611	46.67%
Fourth (Verb)	1000	1309	659	50.34%
Fourth (Adj.)	1000	1309	695	53.09%
Majority(1,3,4)	**1000**	**1309**	**1011**	**77.23%**

*Third(Adj) means We have only considered the Adjective Context words of a word.
Third(Noun) means We have only considered the Noun context words of a word.
Third(Verb) means We have only considered the Verb context words of a word.
Similarly for Fourth(Noun), Fourth(Adj), Fourth(Verb).
Majority(1,3,4) means We have run First, Third and Fourth measure on a word pair and taken the majority decision which gives the highest accuracy.

We have Crossed the Accuracy reported by WeedsPrec which was 63%, which in our case is 77.23%

6 Future Work:

In Future Work, We intend to use the Word Vector to apply supervised model on the dataset.

7 References:

- Scaling to Large3 Data: An efficient and effective method to compute Distributional Thesauri. Martin Riedl and Chris Biemann
- Chasing Hypernyms in Vector Spaces with Entropy . Enrico Santus ,Alessandro Lenci, Qin Lu , Sabine Schulte im Walde .
- Distributed Distributional Similarities of Google Books over the Centuries. Martin Riedl, Richard Steuer and Chris Biemann
- BLESS Dataset. Broni and Lenci
- Unsupervised Hypernym Detection by Distributional Inclusion Vector Embedding.Haw-Shiuan Chang1 , ZiYun Wang2 , Luke Vilnis1 , Andrew McCallum1
- Learning Hypernymy over Word Embeddings. Neha Nayak