B0TB2151

Graduation Thesis

Learning semantic relationships between words in distributional semantics

Jisoo cha

March 2, 2015

Department of Information and Intelligent Systems
Tohoku University

A Graduation Thesis
submitted to Department of Information and Intelligent Systems,
Tohoku University
in partial fulfillment of the requirements for the degree of
BACHELOR of ENGINEERING

Jisoo cha

Thesis Committee:

Professor Kentaro Inui	(Supervisor)
Associate Professor Naoaki Okazaki	(Co-supervisor)
Research Assistant Professor Yuichiroh Matsubayashi	(Co-supervisor)

Learning semantic relationships between words in distributional semantics*

Jisoo cha

Abstract

Many works on extracting hyponymy relationships between words focus on noun pairs, such as *animal* and *dog*. Due to its difficulty, identifying hyponym pairs of verbs has been a big challenge. Yet, the task is vital in various natural language processing tasks like inference and paraphrase. In this paper, I aim to identify hyponymy relations between verb pairs, such as *acquire* and *purchase*. By clustering a verb's direct objects and representing them with normal distribution, given verb pairs were effectively compared. The result shows a high potential of the novel method in hyponym identification of verb pairs.

Keywords:

Distributional semantics, Distributional inclusion hypothesis

*Graduation Thesis, Department of Information and Intelligent Systems, Tohoku University, B0TB2151, March 2, 2015.

Contents

1 Introduction

Distributional semantic model enables us to plot words into a semantic vector space. One of the biggest benefits of this model is that we can calculate the cosine similarity between words. The model extracts words that represent most similar meanings. Extracted words are somehow related, but the type of relationship is not shown by the cosine similarity. For example, *animal* and *cat* are both related to *dog*, but the former one is a hypernym of *dog* while the latter is one of the hyponyms of *animal*, *dog*'s hypernym. Thus, a limitation of cosine similarity lies on the fact that it cannot measure asymmetric relationships such as hyponymy relation.

Yet, identifying hyponymy relations is crucial in many natural language processing (NLP) tasks based on inference. *A dog is running* leads us to infer *An animal is moving*. There are two kinds of hyponymy relations in this inference: one is of noun pairs (*animal* and *dog*), and the other one is of verb pairs (*move*, *run*). Since to *run* is to *move* in a particular manner, the meaning of *moving* includes the meaning of *running*[1]. While there are several works focused on identifying hyponymy relations of noun pairs [2, 3, 4, 5, 8], works on verb pairs are relatively few [12, 13]. However, hyponymy among verbs is vital information in NLP applcations, such as paraphrase and textual entailment.

This paper aims to improve performance in identification of hyponymy relations on verbs with a new method. To be more specific, the paper focuses on direct objects of hypernym pairs of verbs. Consider the following example:

(1) Direct objects of acquire and purchase in sentences.

a. I acquired a <u>skill</u>.
b. I acquired <u>knowledge</u>.
c. I acquired a <u>property</u>.
d. * I purchased a <u>skill</u>.
e. * I purchased <u>knowledge</u>.
f. I purchased a <u>property</u>.

[1] According to a definition provided by Fellbaume [1, p.79], the hyponymy relation between two verbs can be expressed by the formula *To V1 is to V2 in some particular manner*. That means, hyponyms can be related to their superordinates along many semantic dimensions.

The meaning of a verb *acquire* includes the meaning of *purchase*, as *acquire* and *purchase* share a meaning of 'acquire by means of a financial transaction' and *acquire* has further meaning like 'gain knowledge or skills'. The relations are well represented in their direct objects; while *knowledge* and *skill* can be a direct object of *acquire*, but not of *purchase*, *land* or *property* can be a direct object of both verbs. This approach partly rely on **Distributional Inclusion Hypothesis**, which represents: if the meaning of word v includes the meaning of word w, all the syntactic-based features of v are expected to appear with w [5]. As a direct object is one of syntactic features of a verb, checking direct objects to define hyponymy relations can be interpreted as Distributional Inclusion Hypothesis.

Based on the fact that points closely located in the vector space tend to represent similar meaning, I represent a group of objects as a normal distribution model. Compared to the existing methods using syntactic features, this method is rather new. Focusing on direct objects only thus help the normal distribution model to explore its possibility in the task. The reason I chose direct objects is due to its importance among all other features. Direct objects is also the most popular syntactic feature in analyzing relations among verbs [1, p.242][12, 13]. Considering other syntactic features, such as a subject for a verb, can be considered as well as direct objects as a future work.

2 Related work

2.1 Hyponymy among general words

Earlier approaches in hypernym identification rely on pattern-based methods. Hearst utilizes patterns to acquire hypernym and suggests an algorithm to discover new patterns [2]. Patrick and Marco suggest a more accurate algorithm for exploiting generic patterns [3]. In pattern-based approaches, patterns like *such as* helps to detect hypernym pair (*lute*, *Bambara ndang*) from a sentence *The bow lute, such as the Bambara ndang, ...*.

Thanks to semantic vector space, several studies suggest unsupervised hypernym identification in the vector space. A few studies suggest several distributional similariity measures [4, 6]. While these studies are focused on symmetric relations, there are studies on asymmetric relations, such as hyponymy, inference or entailment, based on Distributional Inclusion Hypothesis [7, 8, 9, 10]. Lenci and Benotto compares existing directional similarity measures to identify hypernyms [11]. As the hypernymy relation is asymmetric, directional similarity measures are more appropriate for identifying the relation. In the experiment, this paper compares *balPrec*, one of the best measures in textual entailment task [10], with the new method.

2.2 Hyponymy among verbs

One very interesting study on hypernymy among verbs is Erk's study [12, 13]. Erk represents word meanings as *regions* in vector space, which is very similar to the approach of this paper. Yet the two approachs are different in that while Erk's study generates a region of a verb, this paper represent a region of direct objects of a verb. Erk *encodes* hypernym relations to the regional model with the relation between a target verb and its direct objects. This method is again different with this paper's method that directly compares normal distributions of direct objects of two target verbs.

Although its purpose is not hyponymy identification, Kawahara et al.'s study is very interesting. [14]. They clusters each verb's semantic frames with dependency relations to a verb: nsubj, xsubj, dobj, iobj, ccomp, xcomp, prep_*, to induce semantic frames of each verb. Since the clustered data shows very clear and comprehensive frames, this paper used these semantic frames, instead of clustering direct objects on its own.

3 Method

3.1 Overview

The objective of this paper is to judge if given verbs are in the hyponymy relation. To date, various methods have been developed and introduced to measure directional similarity including hyponym and entailment. Most of these measures assess the relative amount of features common in two vectors compared to the whole set of features within these vectors [10]. One example of directional similarity measures is $balPrec$:

$$balPrec(u \Rightarrow v) = \sqrt{LIN(u,v) \cdot WeedsPrec(u \Rightarrow v)}$$

$balPrec$ is a highly competitive among such measures. It is a geometrical avarage of a symmetric measure LIN and an asymmetric measure $WeedsPrec$, designed to balance the two measures. $LIN(u,v)$ and $WeedsPrec$ are difined as

$$LIN(u,v) = \frac{\sum_{f \in F_u \cap F_v} [w_u(f) + w_v(f)]}{\sum_{f \in F_u} w_u(f) + \sum_{f \in F_v} w_v(f)}$$

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

where F_x is the feature vector of a term x, $w_x(f)$ is the weight of the feature f in that term's vector. LIN measures a similarity between two words, and $WeedsPrec(u \Rightarrow v)$ is a directional measure that quantifies inclusion of the features of the candidate narrower term u by the features of the broader term v.

Although these directional measures are practical in extracting hyponymy relations, they do not consider correlations of features. Since vectors closely located in a semantic vector space have a similar meaning, a group of similar vectors can be regarded as a probability distribution. There are two benefits of this approach. First, the noise coming from sparse data or difference of corpus for training will be eased, as features represents a probability model rather than a point in a vector space. Second, building a probability model itself is meaningful, as it gives a comprehensive and quick idea of the use of a target word, rather than having a group of pure features.

In this section, 3 steps of the method is described: (1) clustering direct objects of a target verb and building a probability model, (2) calculating the distance among clusters by Bhattacharyya distance, and (3) calculating scores of the hyponymy relation between two verbs.

3.2 Step 1. Clustering and a Probability Model

This paper focuses on direct objects to see hyponymy relations on verbs. First, all of direct objects appeared with each verb is extracted from a corpus. In order to compare these objects at step 2, I first cluster them as objects with similar meaning come together and contribute to make a probability model. Among a wide range of existing clustering methods, Gaussian Mixture Model (GMM) was adopted, as GMM returns the most comprehensive clustering results and is easy to build a group of normal distribution models.

The number of clusters ($n\_clusters$) of direct objects rely on the target verb's number of meaning ($n\_synsets$) and the number of direct objects ($n\_objects$). Thus, the number of clusters is defined as

$$n\_clusters = \sqrt{\ln(n\_synsets)\ln(n\_objects)}$$

with the minimum value of 1 and the maximum value of 6[2].

In the evaluation, however, semantic frames from Kawahara et al.'s study [14] were used for the clusters, as GMM requires so enourmous time that it is not appropriate for a big scale of tests. These frames were clustered based on predicate-argument structures of a verb, using English Gigaword corpus.

Direct objects in each cluster then used for generating a multivariate normal distribution model, on Pre-trained Google Word2Vec, a 300-dimensional semantic vector space [15]. A multivariate normal distribution model is represented by these formulas:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$$

[2]The maximum value was decided according to the running time: the setting of more than 6 clusters required enourmous time.

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^t$$

where $\boldsymbol{\mu}$ is a the mean of the distribution, Σ is the covariance matrix, and $\boldsymbol{x}$ is a vector of all direct object in each cluster. By using these parameters, the probability density function (pdf) is defined as

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^t\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right) \tag{1}$$

where $|\Sigma|$ is the determinant of Σ. This equation (1) is used for the evaluation. However, in some cases, it was impossible to calculate the determinant, due to data sparseness compared to 300 dimensions. To handle this problem, the covariance Σ was recalculated by Shrunk Covariance, using Scikit, a Python tool.

3.3 Step 2. Distance between Clusters

To calculate the distance between clusters, the **Bhattacharyya distance** is used. the Bhattacharyya distance measures the similarity of two discrete or continuous probability distributions. For multivariate normal distributions, the Bhattacharyya distance D_B is defined as

$$D_B = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\ln\left(\frac{|\Sigma|}{\sqrt{|\Sigma_1||\Sigma_2|}}\right)$$

where $\boldsymbol{\mu}_i$ and Σ_i are the means and covariances of the distributions, and

$$\Sigma = \frac{\Sigma_1 + \Sigma_2}{2}$$

3.4 Step 3. Hyponymy Score

The task in step 3 is to compare given verbs by their clusters of direct objects. Consider a verb pair of hypernym and hyponym. For the hypernym verb that has a broader meaning, direct objects must be distributed broadly than a hyponym that has a narrower meaning. This idea gives a formula:

$$Score(A \supset B) = \frac{1}{m}\sum_{j=1}^{m}\min_{i=1,\ldots,n}(D_B(C_{Ai}, C_{bj})) \tag{2}$$

where n and m are the number of clusters in verb A and B, C_{Ai} is a cluster of verb A, and C_{Bi} is a cluster of verb B. This equation of score increases when the clusters of A is broader than the clusters of B, or two groups of clusters have no relation. The score decreases when the clusters of A and the clusters of B are closely related and the clusters of A generally includes every cluster of B. Note that the score is not symmetric: even if A and B is highly related, when B has broader clusters, some of these clusters returns higher D_B, thus the score increases.

4 Experiment

This section evaluates the performance of the model described in Section 3. I test two main challenges on the model: (1) how well does the model predict the occurrence of direct objects of a target verb (experiment 1), and (2) how well does the model judge the hyponymy relation between verbs (experiment 2).

4.1 Experiment 1: Verb and Object Pairs

4.1.1 Experiment Setting

By the method described in Section 3, a group of clusters are acquired on each verb. Before testing its performance on hyponymy identification, the accuracy of the model itself should be tested. To test its validity, positive pairs and negative pairs of (verb, direct object) are given to the model. 1,500 positive pairs and 1,500 negative pairs are obtained from PukWaC corpus. Positive pairs consist of only pairs that did not appear in direct object sets of the target verb (e.g. (*forget*, *flight*) or (*mail*, *agreement*)). Negative pairs are randomly generated from direct object sets of another verb, that do not appear in the model (e.g. (*divorce*, *salsa*) or (*afford*, *darkness*)). For a given pair, the probability density function (pdf) of each cluster of a target verb was calculated, through the equation (1), and then the maximum of the pdf value is returned. Half of the whole pairs, 750 positive and 750 negative pairs, are used as a development set, to decide a threshold of pdf, and the rest half was used as a test set with the thresold.

4.1.2 Experiment Result

Table 1 presents the result of experiment 1. D_B refers to the method calculating pdf described at Step 1 in Section 3. *Cosine* measures the cosine similarity between a verb and an object, in the Pre-trained Google Word2Vec, the 300-dimensional vector space. *Cosine* is used as a baseline to evaluate the model D_B. As can be seen from the table 1, D_B effectively predicts the occurance of (verb, direct object) pairs, and beat the baseline. From this, we can see that building a regional model on a semantic vector space is a valid idea.

Method	F1	Precision	Recall	Threshold
D_B	**0.6922**	0.5988	0.8200	-1106.00 (log value)
Cosine	0.6473	0.5185	0.8613	0.0007

Table 1. Result of experiment 1: The model D_B beats the baseline *Cosine*.

4.2 Experiment 2: Verb Hypernym Pairs

4.2.1 Experiment Setting

In this experiment, the model predicts whether a given verb pair of (hypernym, hyponym) is in the hyponymy relation. 487 positive and 487 negative pairs were used to test the performance of the model. Positive pairs were extracted from WordNet hyponym relations (e.g. (*make*, *generate*) or (*give*, *donate*)), and negative pairs are random verbs that are not in the hyponymy relation (e.g. (*eliminate*, *remember*) or (*run*, *possess*). The *Score*(verb A $\supset$ verb B) is then calculated with the equation (2).

There are two ways of hyponym identification, *simple* and *complex*. *simple* way is to see only $Score(A \supset B)$, as shown below.

1. If $Score(A \supset B) \leq$ Threshold, then verb A is a hypernym of verb B.
2. If $Score(A \supset B) >$ Threshold, then verb A is not a hypernym of verb B.

complex way considers both $Score(A \supset B)$ and $Score(B \supset A)$:

1. If $Score(A \supset B) \leq$ Threshold and $Score(B \supset A) \leq$ Threshold, then verb A and verb B are synonyms.
2. If $Score(A \supset B) \leq$ Threshold and $Score(B \supset A) >$ Threshold, then verb A is a hypernym of verb B.
3. If $Score(A \supset B) >$ Threshold and $Score(B \supset A) \leq$ Threshold, then verb B is a hypernym of verb A.
4. If $Score(A \supset B) >$ Threshold and $Score(B \supset A) >$ Threshold, then verb A and verb B have no relation.

The experiment was conducted in both ways.

Method	F1	Precision	Recall	Threshold
$balPrec$	0.7407	0.6881	0.8021	0.23
D_B	0.6778	0.5200	0.9733	72.75
D_B_freq	0.6845	0.5327	0.9572	68.25
$Cosine$	**0.7530**	0.6826	0.8396	0.475
$Cosine\_freq$	0.7371	0.6569	0.8396	0.454

Table 2. Result of experiment 2 with *simple* way: The model D_B and D_B*_freq* does not perform as well as the baseline *balPrec*. Instead, *Cosine* shows the best performance.

Among the total of 974 pairs, 300 positive pairs and 300 negative pairs are used as a development set, to decide the threshold. The rest of 187 positive pairs and 187 negative pairs was then tested on the model.

4.2.2 Experiment Result

Table 2 and Table 3 shows the results of experiment 2, in a *simple* way and in a *complex* way, respectively. Although the model does not perform very well in the first experiment, compared to the baseline and the cosine measure, D_B and D_B_freq outperformes the others in the second experiment. This result may be explained by the fact that the model was designed in a way to compare two scores of $Score(A \supset B)$ and $Score(B \supset A)$, not alone.

Figure 1 represents a good exmaple and a bad example of clustering results. For the positive pair (*choose*, *elect*), the model successfully extracts the two most closest clusters that include direct objects appear in both verbs. Nouns in those clusters are related to a political election. On the other hand, *choose*[0] shows the nouns that only appear with choose, increasing the distance between itself and *elect*[4]. For the negative pair (*frustrate*, *beg*), scores show randomely generated two verbs are close to each other, which is undesirable. The main reason is that clusters containing pronouns like him, or them are regarded rather close in the model. Yet, those pronouns are hardly representing the characteristic of the target verb.

A good example of clustering of a pair (*choose*, *elect*)

$n\_clusters$ of *choose* $= 13$, $n\_clusters$ of *elect* $= 6$
$Score(choose \supset elect) = 31.9$, $Score(elect \supset choose) = 50.2$

- The two most closest clusters ($D_B = 23.4$):
  choose[4]: president, leader, government, minister, parliament, head, ...
  elect[4]: leader, speaker, head, chairman, leadership, board, bishop, ...
- The two most furthest clusters ($D_B = 135.2$):
  choose[0]: reason, fact, him, occasion, chance, week, it, concern, ...
  elect[4]: leader, speaker, head, chairman, leadership, board, bishop, ...

A bad example of clustering of a pair (*frustrate*, *beg*)

$n\_clusters$ of *frustrate* $= 6$, $n\_clusters$ of *beg* $= 3$
$Score(frustrate \supset beg) = 27.9$, $Score(beg \supset frustrate) = 40.9$

- The two most closest clusters ($D_B = 21.5$):
  $frustrate[1]$: **him**, people, group, **them**, **you**, **us**, **her**, audience, public, ...
  beg[2]: **you**, **them**, **me**, one, mom, daughter, money, boss, audience, ...
- The two most furthest clusters ($D_B = 87.7$):
  $frustrate[4]$: will, user, goal, investor, consumer, viewer, interest, ...
  beg[2]: you, them, me, one, mom, daughter, money, boss, audience, ...

Figure 1: A good exmaple and a bad example of clustering results. $A[i]$ represents ith cluster of a verb A.

Method	F1	Precision	Recall	Threshold
$balPrec$	0.4538	0.8082	0.3155	0.248
D_B	**0.5298**	0.6957	0.4278	54.23
D_B_freq	**0.5350**	0.6614	0.4492	54.6
$Cosine$	0.4502	0.7262	0.3262	0.541
$Cosine\_freq$	0.4502	0.7262	0.3262	0.534

Table 3. Result of experiment 2 with *complex* way: The model D_B and D_B*_freq* beats the baseline *balPrec*.

5 Conclusion

This paper was undertaken to design an appropriate model for verb hyponymy identification. The research has proposed a novel method to see the inclusion relationship between verbs. First, the new model clusters direct objects of two given verb pairs. Then, it compares the clusters by calculating a distance between each cluster pairs. Finally, two scores are generated and used for the identification.

In the experiments, two challanges were tested: (1) whether the model can predict given verb and object pairs, and (2) whether the model can identify hyponymy relations. The model successfully showed its possibility in the evaluation.

Future research needs to be carried out in order to overcome limitations of the model and obtain a better performance. One possible topic is to explore the model with a Gaussian Mixture Model without using pre-obtained frames. Another possible topic is to set the model with more semantic features as well as direct objects.

The result of the evalution on the new model shows that it has a high potential power in verb hyponymy identification, after some modifications. A key strength of the present study was that it generates a probability model on the semantic vector space. Thus, this study also provides a framework for the exploration of regional expression in a vector space.

Acknowledgements

I would like to express the deepest appreciation to Professor Inui Kentaro, Associate Professor Naoki Okazaki, and Research Assistant Professor Yuichiroh Matsubayashi for guiding me and giving an inspiration of this research. Without their supervision and constant help, this thesis would not have been possible. I would like to thank the laboratory members, for giving me valuable comments on my research and having discussions. I also thank Professor Daisuke Kawahara for permission to use his framework in this research.

References

[1] Miller, George, and Christiane Fellbaume. 1998. Wordnet: An electronic lexical database. *MIT Press Cambridge.*

[2] Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*: 539—545.

[3] Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the COLING-ACL 2006*, Sydney, Australia: 113—120

[4] Lee, L. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, MD, USA, pp. 25—32

[5] Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of ACL*: 107—114.

[6] Weeds, J., and Weir, D. 2003. A general framework for distributional similarity. In *Proceedings of EMNLP, Sapporo, Japan.*

[7] Bhagat, R., Pantel, P., and Hovy, E. 2007. LEDIR: an unsupervised algorithm for learning directionality of inference rules. In *Proceedings of EMNLP-CoNLL, Prague, Czech Republic.*

[8] Idan Szpektor and Ido Dagan. 2008. Learning Entailment Rules for Unary Templates. In *Proceedings of COLING 2008, Manchester, UK*: 849—856.

[9] Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the EACL 2009 Workshop on GEMS: GEometrical Models of Natural Language Semantics, Athens, Greece*: 112—119.

[10] Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359—389.

[11] Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *Proceedings of Association for Computational Linguistics 2012*: 75—79.

[12] Katrin Erk. 2009. Supporting inferences in semantic space: representing words as regions. In *Proceedings of the 8th International Conference on Computational Semantics, Tilburg, January*: 104—115.

[13] Katrin Erk. 2009. Representing words as regions in vector space. In *Proceedings of the Thirteenth Con- ference on Computational Natural Language Learning (CoNLL), Boulder, Colorado*: 57—65.

[14] Daisuke Kawahara, Daniel W. Peterson, and Martha Palmer. A Step-wise Usage-based Method for inducing Polysemy-aware Verb Classes. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*: 1030—1040.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS, 2013.*