Alternate Equivalent Substitutes: Recognition of Synonyms Using Word Vectors

Tri Dao, Sam Keller, Alborz Bejnood
Stanford University – CS 229: Machine Learning

December 12, 2013

1 Abstract

The task of computationally learning the linguistic application of a word requires a robust analysis of both the word's semantic presence and its independent characteristics. Word vectors are learned, functional representation of words which can then be analyzed for linguistic regularities. We examine whether the synonyms of a word can be recognized from their word vectors by training a neural network on a large corpus of text, then implementing k-means clustering to check whether synonyms have a statistically significant word vector similarity. We illustrate the output using a simplified version of the vectors generated by principal component analysis. Results suggest that increasing vector length, which improves syntactic and semantic accuracy when performing comparisons of words, negatively correlates with synonym recognition.

2 Introduction

Effective representations of lexical tokens are dependent on a perceptive consideration of the multiple characteristics that define words. These characteristics range from more concrete attributes (including but not limited to the definition, part of speech, and tense) to the semantic relations of the word resulting from the context in which the word is used. For example, consider the polysemous word 'cold.' A comprehensive understanding would include the definitions of all the different senses of the word (*of or at a relatively low temperature*, *lacking affection or warmth, a common viral infection*), the part of speech (*adjective, noun, adverb*), along with some knowledge of the surrounding text (*he stopped cold*).

The system we used represents these traits using **word vectors:** real vectors that represent the word in n-dimensional space. The vectors are learned using a **feed-forward neural network,** a classification algorithm that passes an input through several layers of processing units until reaching an output layer. We used the open-source tool Word2Vec to generate these word vectors, which in turn were used to investigate the potential of utilizing a combination of machine learning techniques with natural language processing ideas in order to recognize synonomous words.

3 Data

We obtained and processed the following data:

- Text corpus
- List of words
- List of corresponding synonyms
- Text representation framework

In order to effectively incorporate a diverse set of words, we used a corpus text from the Wesbury Lab Wikipedia corpus, comprising over 2 million documents and approximately a billion words. We extracted synonyms from WordNet (a large lexical database for the English language) by using a provided file consisting of 5000 of the most commonly used English words, removing lexically meaningless tokens such as 'the' and 'a', and cross checking the remaining terms with the 117,000 sets of synonyms in the WordNet database in order to find the sets containing each word.

4 Models

We generated word vectors based on the following two models: `Bag-of-Words` and `Skip-gram`.

4.1 Bag of Words Model

The `Bag-Of-Words` text representation model consists of treating a sentence as a map from words to the frequency of that word in a sentence. Data is not stored about the grammatical nature of the sentence or word order, thus information about the semantic meanings of words is lost. However, this is a useful baseline approach to determine general similarities between words. The architecture is illustrated in Figure 1 (taken from [1]).

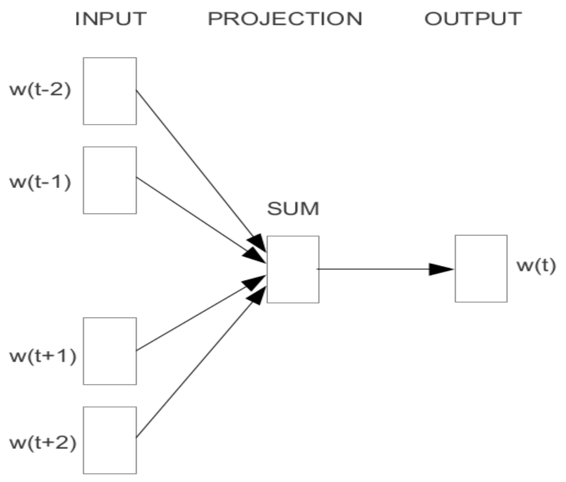

Figure 1: Bag of words model. Predicts current word based on surrounding words.

4.2 Skip-gram Model

The idea behind the `Skip-gram model` is an extension of the idea used in the single n-gram model: we want to look at not only the set of adjacent words, but also sets of words where some (up to the order of the skip gram) are skipped. The parameter that controls how many words the model can skip is called the window parameter. In this paper, we are primarily interested in the Skip-gram model with the window parameter of 2, 4, 6, and 10. The architecture is illustrated in Figure 2 (taken from [1]).

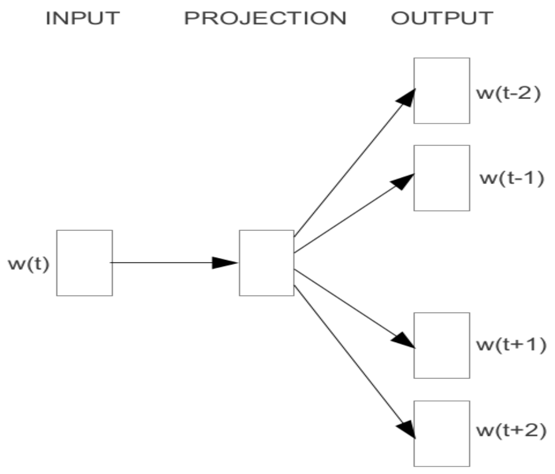

Figure 2: Skip-gram model. Predicts surrounding words based on current word.

5 k-means clustering

The clusters were calculated using a reduced corpus of more common words, with initial values randomly selected from that corpus.

`k-means Clustering`

1. Initialization: Choose a random word
2. Iteration: Assign each point to nearest mean
3. Move mean to the center of the cluster

We implemented our version of the k-means clustering algorithm (as written above) and used it to test synonym recognition as described below.

```
Synonym Recognition Algorithm
```

1. Run k-means clustering on small set of words (4000)
2. Assign each word in large corpus to a cluster
3. For each word in small corpus, compute:
    - percentage of synonyms in same cluster
    - ratio of percentage to expected percentage

To calculate the ratio of percentage to expected percentage, for every word vector we compute the probability of one of its synonyms being assigned to the same cluster as itself, and then find the average probability by summing over all words.

6 Results

We first computed the semantic accuracy and syntactic accuracy of the word vectors obtained using the internal testing tool that came with Word2Vec, which consists of 30000 analogy questions. For example, a semantic question may have the form of `man woman king queen`. If vec(*man*) − vec(*woman*) + vec(*king*) is closest to vec(*queen*), then the word vector model is consider to be correct. A syntactic question may have the form `possible possibly quick quickly`. The accuracy is the percentage of questions that the model get correctly. Figure 3 shows graph of overall accuracy (the average of semantic accuracy and syntactic accuracy) for different models: Bag of Words and Skip-gram with window parameter of 2, 4, 6, and 10. This measures how much the word vectors can capture the meaning and grammatical information of the words they represent.

The k-means clustering results are given for Bag of Words model in Table 1 and Skip-gram (with window parameter of 2) in Table 2. This is the ratio between the percentage of a synonym

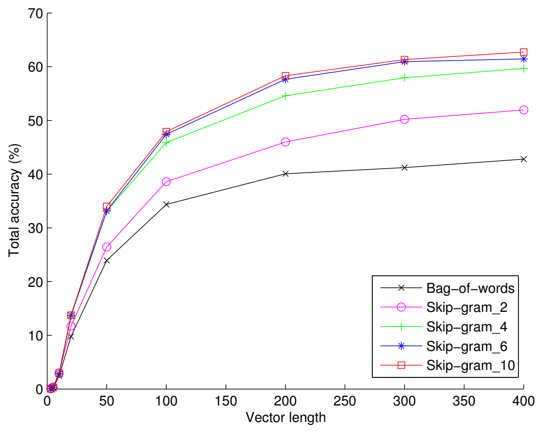

Figure 3: Overall accuracy of different models

of a word being in the same cluster as the word itself, compared to the percentage of any randomly chosen word to be in the same cluster as the word itself. Larger number indicates better synonym recognition.

Length	Number of Clusters			
	2	4	6	8
3	1.159	1.291	1.310	1.301
5	1.145	1.316	1.471	1.491
10	1.119	1.299	1.402	1.536
20	1.093	1.207	1.396	1.502
50	1.085	1.231	1.319	1.464
100	1.019	1.160	1.265	1.241
200	0.916	1.043	1.037	1.012
300	0.879	0.968	0.931	0.931
400	0.943	0.944	0.829	0.902

Table 1: k-means clutering result for Bag-Of-Words model

We also computed the average distance between a word and its synonyms. This distance is then normalized by dividing by the distance between that word and any randomly chosen words. The graph of this normalized distance is plotted in Figure 4. Lower distance means better synonym recognition.

Length	Number of Clusters			
	2	4	6	8
3	1.175	1.339	1.337	1.488
5	1.074	1.353	1.542	1.548
10	1.125	1.343	1.508	1.575
20	1.156	1.303	1.383	1.613
50	1.098	1.180	1.390	1.554
100	1.051	1.148	1.257	1.368
200	1.039	1.025	1.048	1.082
300	0.957	0.963	0.934	0.948
400	0.975	0.870	0.815	0.851

Table 2: k-means clutering result for Skip-gram model (window of 2)

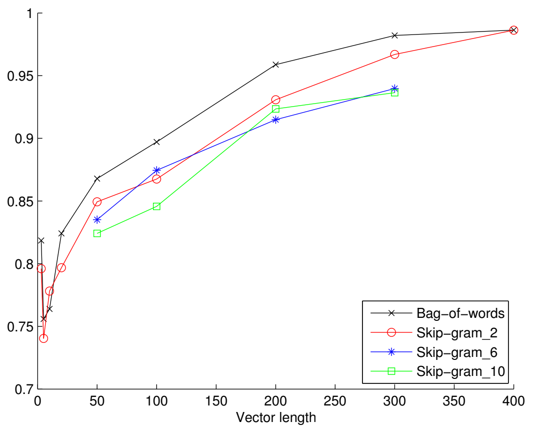

Figure 4: Average distance between a word and its synonyms

7 Analysis

From the graph is Figure 3, it is clear that increasing the vector length yields significantly higher accuracy. Larger vector length means the word vectors are able to capture more semantic and syntactic information of the words they represent.

To investigate the data in Table 1 and Table 2, we should note that a larger ratio implies a larger percentage of synonyms that were mapped to the same cluster as their original word, which implies better synonym recognition. Coupled with the plot of the normalized average distance between a word and its synonyms, we see a reverse trend: smaller vector lengths significantly improves the probability of detecting synonyms. These results are statistically significant since, for example, using the Skip-gram model with window of 2 words, the synonyms of a given word are about 1.6 times more likely to be in its cluster as a randomly chosen word.

However, the length of the vector representation cannot be too small. Any length less than 10 decreses the likelihood of detecting synonyms. The best vector length varies from 10 to 20.

An illustration of the clustering is shown with an application of the PCA algorithm which maps large-dimensional vectors to 2-D space. Figure 5 corresponds to a Skip-gram model with vector length of 200. The words in red are "beautiful" and its synonyms, and the words in black are several randomly selected adjectives. We see that synonymous words do not necessarily cluster together. Figure 6 shows the same Skip-gram model with vector length of 20, mapped for "cold" and its synonyms. This time, synonymous words cluster closer together.

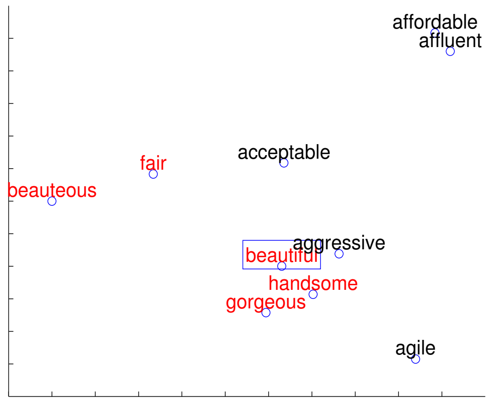

Figure 5: "beautiful" and its synonyms, along with randomly selected adjectives

A longer vector length means having more fea-

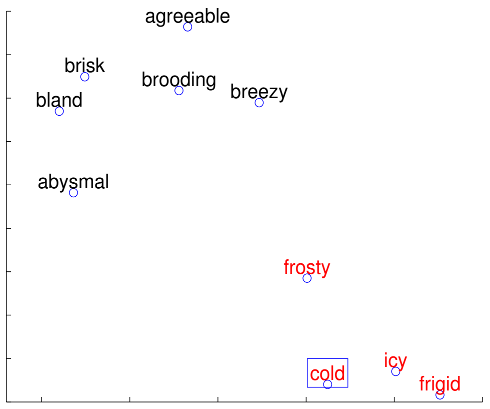

Figure 6: "cold" and its synonyms, along with randomly selected adjectives

tures, some of which may have no bearing on the concept denoted by the word. Instead, they may capture the syntactic information or the context of how the word is used. This is in fact how the neural network is trained: it has no access to the meaning of the words, only the context of how it is used. When we then use this high-dimensional data to compute distance between a word and its synonyms, the conceptually irrelevant portion of the representation causes the distance to be much larger. When we limit the number of features in the internal representations, more of the vector representation contains conceptually relevant information. As such, the distance between a word and its synonyms might be smaller.

Using word vector representations can give a meaningful result since the synonyms of a given words are likely to be in its cluster. However, this is not necessarily a good enough metric to detect synonyms. These cluters are large and contain hundreds of words. If we only rely on the clutering data or distance data, we cannot conclude definitely that two words are synonyms. However, since this is a statistically significant result, it can be used to improve synonyms recognition when paired with other methods, for example, syntactic and semantic analysis.

8 Future Work

These results suggest several extensions to generating a more effective classification system. We propose the following two alternatives which are likely to improve synonym detection using clustering and distance computation of word vectors:

1. Train the neural network with a large vector size to increase syntactic accuracy using word vectors. However, maintain another model with a small vector size, preferably from 10 to 20, which yield a much better probability of synonyms of a given word having a vector representations close to the representation of the word itself.

2. Traing the neural network with a large vector size as before. However, during training, add another class of parameters: relative weights for each dimention of the vector representation (i.e. for each feature). Train the neural network to optimize these weights so as to minimize distance between words and its synonyms, as given from a lexical database. Since we are not losing any feature, the semantic and syntactic accuracy of the model is maintained. However, synonyms recognition will be much improved.

References

[1] Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space.* In Proceedings of Workshop at ICLR, 2013.

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

[3] J. Turian and L. Ratinov and Y. Bengio, *Word representations, a simple and general method for semi-supervised learning.* ACL, 2010.