Alternate Equivalent Substitutes: Examining The Recognition of Synonyms Using Word Vectors

Alborz Bejnood, Tri Dao, Sam Keller
Stanford University
Computer Science 224N: Natural Language Processing

December 6, 2013

1 Abstract

The task of computationally learning the linguistic application of a word requires a robust analysis of both the word's semantic prescence and its independent characteristics. Word vectors, which are learned representations of words, offer a 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 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 vector length varies inversely with the ability to recognize synonyms, while being directly related to semantic accuracy.

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, e.g. *of or at a relatively low temperature*, *lacking affection or warmth*, *a common viral infection* and the part of speech *adjective, noun, adverb*, along with some knowledge of the surrounding text *she quit cold turkey*, *he stopped cold*.

The system we used represents these traits to computers using **word vectors:** vectors consisting of numeric elements that represent the word in n-dimensional space (where n is the size of the vector). Word vectors can be learned through the use of **feed-forward neural networks:** a classification algorithm that passes an input through several layers of processing units (with no feedback from higher layers to lower ones, meaning that as suggested by the name, information is always passed forward) until reaching an output layer. The online tool Word2Vec was used for the generation of these word vectors, which in turn were used to investigate the potential of utilizing a combination of ma-

chine learning techniques with natural language processing ideas in order to recognize synonomous words.

3 Data

The following data had to be obtained and processed:

- 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. Synonyms were extracted from WordNet 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 corpus 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 `Skipgram`.

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 dictionary. Data is not stored about the grammatical nature of the sentence, or the word order, which discounts semantic meanings of words. However, this remains a useful baseline approach to determine general similarities to words. The architechture is illustrated in Figure 1.

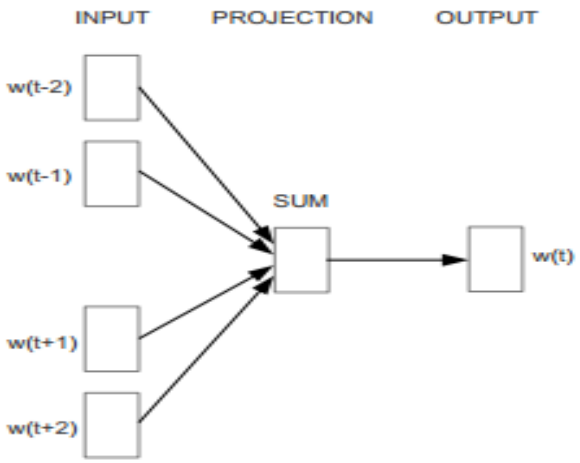

Figure 1: Bag of words model. Predicts current word based on surrounding words.

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 adject words, but also sets of words where some (up to the order of the skip gram) are skipped. The architecture is illustrated in Figure 2.

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`

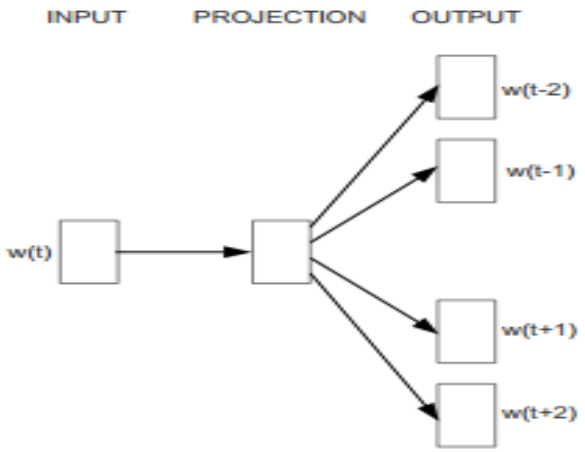

Figure 2: Skip-gram model. Predicts surrounding words based on current word.

1. Initialization: Choose a random
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), using it to test synonym recognition as described below.

`Synonym Recognition Algorithm`

- Run k-means clustering on small set of words (4000)
- for 'common word' in synonyms.txt:
- assign each word in large corpus to cluster
- for each synonym of common word:
- compute percentage of synonyms in same cluster
- compute 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 it, and then find the average probability by summing over all words.

6 Results

The results are given for Bag of Words model in Table 1, Skip gram in Table 2, for different parameter values.

	Num Clusters				
Vector Length	2	4	6	8	
3	1.159	1.291	1.310	1.301	0.2466
5	1.145	1.316	1.471	1.491	0.2466
10	1.119	1.299	1.402	1.536	0.2466
20	1.093	1.207	1.396	1.502	0.2466
50	1.085	1.231	1.319	1.464	0.2466
100	1.019	1.160	1.265	1.241	0.2466
200	0.916	1.043	1.037	1.012	0.2466
300	0.879	0.968	0.931	0.931	0.2466
400	0.943	0.944	0.829	0.902	1.0

Table 1: Vector length as function of clusters

	Num Clusters			
Vector Length	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: Vector length as function of clusters

7 Analysis

To investigate the data in Table 1, we should note that a larger ratio implies a larger per-

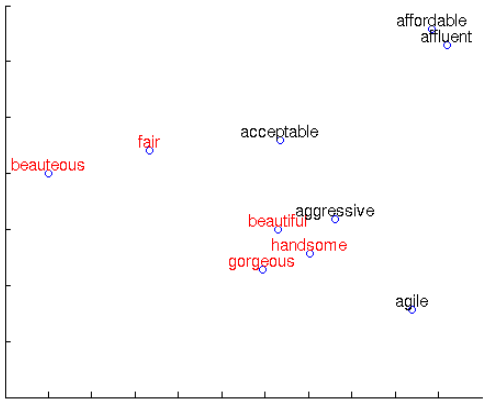

Figure 3: "beautiful" and its synonyms, along with randomly selected adjectives

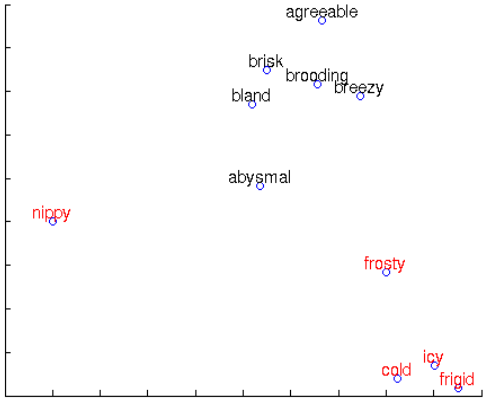

Figure 4: "cold" and its synonyms, along with randomly selected adjectives

centage of synonyms that were mapped to the same cluster as their original word, which implies better synonym recognition. Moreover, from the plot of the clustering ratio and the normalize average distance between a word and its synonyms, we see a somewhat surprising result: 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 3 corresponds to a 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. In contrast, figure 4 corresponds to a model with vector length of 200. The words in red are "cold" and its synonyms, and the words in black are several randomly selected adjectives. This is an example of synonomous words get clustered more closely in the cases corresponding to the smaller vector sizes.

A longer vector length means having more features. Some of the features may be sematically irrelevant, that is, has no bearing to the meaning of 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: the neural network 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 semantically irrelevant portion of the representation causes the distance to be much larger. When we limit the num of features in the internal representations, more of the vector representation contains sematically relevant information. As such, the dis-

tance 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, we should note that this is not necessarily a good enough metric to detect synonyms. These cluters are large and contain hundreads 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 hybrid version which is likely to improve synonym detection using clustering and distance run of word vectors:

1. Use word2vec with a large vector size to increase syntactic accuracy using word vectors.

2. Maintain another model with a small vector size, probably 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.

9 References

1. J. Turian and L. Ratinov and Y. Bengio, Word representations. *A simple and general method for semi-supervised learning.* ACL, 2010.

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

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