Comparison of Word2vec and GloVe Word Embeddings

Corey Shih

coreys2@illinois.edu

Introduction:

Common natural language processing tasks often treat words as simple and independent units to be used as features for NLP algorithms; one such example is the ever-prevalent bag-of-words model. While such techniques are simple and oftentimes sufficient for the problem at hand, they make no attempt to model semantic and syntactic similarities between words, which may be necessary for more complex NLP applications. Word embeddings are a language modeling technique that attempt to map words to vector representations, allowing their semantic and syntactic similarities to be easily compared. In general, words with similar meanings will have similar representations in the vector space. Additionally, word embeddings support analysis of analogies or word algebra; that is, taking the embeddings of the words *king*, *queen*, *man*, and *woman*, subtracting *man* from *king* and adding *woman* would result in a vector very similar to that of *queen* [1]. Word embeddings were first introduced in 2013 by researchers at Google led by Tomas Mikolov, who developed word2vec [2]. In 2014, Stanford released a competing word embedding technique known as GloVe, or Global Vectors for Word Representation. These two approaches remain among the most commonly-used today. This review presents a broad overview of both techniques and a comparison of their similarities and differences.

Discussion:

Word2vec

Word2vec utilizes a shallow neural network to compute vector representations of words. The model assumes that words with similar contexts in a corpus have similar meanings, and thus attempts to position words with similar contexts close to each other in vector space. The cosine similarity between vectors is used to quantify similarity in vector space. There are two primary approaches to implementing word2vec: continuous bag of words (CBOW) and continuous skip-gram.

In the CBOW approach, the model attempts to predict a target word given a window of context words around the target word. The set of context words is treated using the bag-of-words model, so information regarding the order of context words is not retained. In the continuous skip-gram approach, the model attempts to predict the surrounding context words of a given target word, assigning higher weights to nearby context words and lower weights to context words further away. This approach is reminiscent of the opposite of the CBOW approach. Analysis conducted

by the authors of word2vec concluded that the CBOW approach trains faster than the skip-gram method, but the skip-gram model is superior when it comes to representing infrequent words in the corpus [3].

GloVe

The primary assumption underlying the GloVe model is that word-word co-occurrences encode some form of meaning of the words in question. For example, the word *ice* co-occurs more frequently with *solid* than *gas*, and *steam* co-occurs with *gas* more frequently than *solid* [1]. Both words co-occur with *water* frequently and *fashion* infrequently. As we humans know, *ice* is the solid form of *water*, steam is the *gas* form of *water*, and both *ice* and *steam* are unrelated to *fashion*. By taking ratios of word-word co-occurrence probabilities for *ice* and *steam*, the model can learn some conceptual information relating *solid* to *ice* and *gas* to *steam*.

The GloVe model is trained using the non-zero entries of the word-word co-occurrence matrix of a corpus. The initial construction of this matrix is computationally expensive, but subsequent training iterations are significantly faster due to the fact that the number of non-zero entries is generally much lower than the number of words in the corpus. In training the model, GloVe attempts to learn word vector representations such that the dot product of two word vectors equals the logarithm of the probability of the co-occurrence of the words. Due to the fact that a logarithm of a ratio is equal to a difference of logarithms, ratios of co-occurrence probabilities of words are encoded as vectors differences between the word vectors. For this reason, GloVe tends to perform well on analogy tasks.

Conclusion:

Word2vec and GloVe are both popular word embedding techniques. While word2vec learns word embeddings by relating a target word to its context, GloVe instead uses word-word co-occurrences to derive word vector representations. While the intuition behind the two approaches is quite different, the two methods seems to work similarly well on a variety of NLP tasks. The word embeddings learned by both models capture semantic and syntactic relations between words, which cannot be obtained from many simpler NLP modelling techniques.

References:

[1] J. Pennington, R. Socher, C.D. Manning. Glove: Global vectors for word representation. EMNLP 2014, 1532-1543.

[2] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient estimation of word representations in vector space. arXiv 2013.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 2013, 3111-3119.