Representing Words as Low-dimensional Vectors: A Comparative Review of Word2Vec, FastText and GloVe

Junzhe Sun (junzhes2)

Introduction

Word embedding is fundamental to natural language processing (NLP). It is one of the first steps in the NLP pipeline, right after data collection and data cleaning. Learning-based word embedding tools are simple in principle yet effective in practice. Understanding these elegantly formulated “shallow-learning” approaches provides insights into more sophisticated deep-learning tools, such as LSTM and Transform networks. The goal of word embedding is to project words in a vocabulary into a low-dimensional, dense vector space, which captures the similarity and analogy between pairs of words. Some of the most popular word embedding techniques include Word2Vec, FastText and GloVe. The goal of this paper is to review these methods in a comparative manner. This review doesn’t cover any mathematical equations (as they are readily available in the original papers), but makes an effort to describe each formulation in plain language that highlights their similarities and differences.

Review of Word Embedding Approaches

Word2Vec

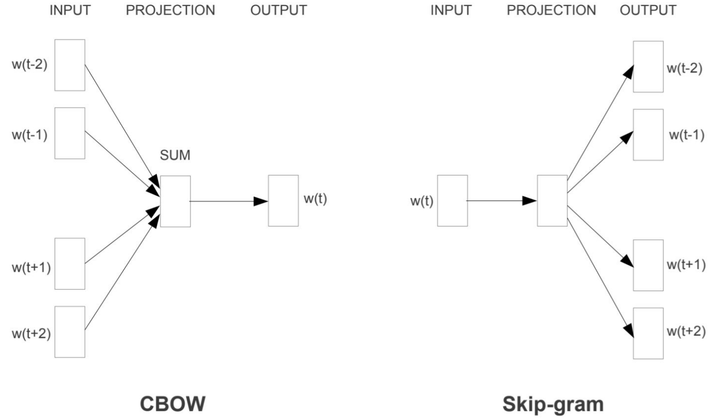

Figure 1. Comparison of CBOW and Skip-gram (Mikolov et al.,2013a).

Word2Vec was introduced in two influential papers by a team of researchers from Google in 2013 (Mikolov et al.,2013a and Mikolov et al., 2013b)). The key idea proposed in Mikolov et al. (2013a) was that instead of training a feedforward neural network language model (NNLM) with a non-linear hidden layer that could potentially represent the data more accurately (Bengio et al., 2003), a simpler, log-linear model was adopted to allow more efficient training on large amounts of data. According to Mikolov et al. (2013a), the two main steps of learning a neural language model includes: (1) learning of continuous word vectors using a simple model; (2) training of N-gram NNLM on top of the learned word vectors. Two log-linear models were introduced, including the continuous bag-of-words (CBOW) model and the continuous skip-gram model. The relationship between CBOW and skip-gram is illustrated in Figure 1. In CBOW, the training task is to predict the current/center word using the context/surrounding words, while in the skip-gram model, the training task is to predict words within a certain range before and after the input “current” word. Both models are shown to be able to effectively train a neural language model that captures word similarities and relationships.

The original formulation will lead to a high computational cost, and a hierarchical softmax was adopted where the vocabulary is represented as a Huffman binary tree. To further improve the quality and training efficiency of Word2Vec, Mikolov et al. (2013b) introduced two influential techniques. The first idea is known as “negative sampling”, and aims to improve the training speed. Negative sampling is based on a simplification of the idea of Noise Contrastive Estimation (Gutmann and Hyvarinen, 2012), and provides an alternative approach to hierarchical softmax. The main idea is that instead of computing all combinations of center and context word vectors, K randomly sampled words from the vocabulary are used as negative examples and binary logistic regression is used instead of a softmax. This could significantly reduce the computational complexity and therefore improve training speed. The second idea is similar to the inverse-document frequency weighting in text retrieval. For very large corpora, common “stop words” such as “in”, “the” and “a” are the most frequent words but do not necessarily contribute to an effective word embedding, while less frequent words could contain more useful information for learning word representations. To re-balance the contribution from rare and frequent words, a simple subsampling was proposed that more frequent words are more likely to be discarded. This mainly improves the equality of the output vectors.

Word2Vec is a single-layer neural network and has a simple objective (predicting a center word given its context, or predicting the context given a word). However, its effectiveness in capturing word relationships has made it one of the most popular techniques for word embedding, providing a suitable input representation for more advanced NLP algorithms.

FastText

FastText is a more recent word embedding framework that is gaining a lot of popularity lately, and it is a close sibling to Word2Vec. Introduced by Bojanowski et al. (2017) from Facebook AI Research, FastText is a method that aims to extend word vectors to capture subword information, instead of representing each word in the vocabulary as distinct vectors and ignore the internal structure of words. In FastText, low-dimensional representations of character n-grams are learned, and words are represented as the sum of the n-gram vectors (similar to the relationship between words and CBOW). This model is able to capture subword information, and allows sharing of such information across words. Several advantages of FastText were reported by the authors. First, nearest neighbors measured by cosine similarity are more accurate using FastText compared with Word2Vec. Second, grammatical variations and compound nouns are easier to model with character n-grams. Finally, rare words and even out-of-vocabulary words can be reliably represented by word vectors produced by FastText. One caveat is that building n-gram dictionaries can be memory intensive, and the solution proposed by the authors is to use a hashing function that maps n-grams to integers of a limited range. Training a FastText word embedding is fast (as its name indicates) and it tends to outperform Word2Vec which does not consider subword information according to the authors.

GloVe

GloVe stands for global vectors, which is yet another popular word-embedding formulation introduced by Pennington et al. (2014). GloVe aims to combine the advantages of global matrix factorization and local context window methods, the latter being the approach adopted by Word2Vec. The development of GloVe was motivated by the observation that global matrix factorization methods efficiently leverage statistical information of words, while local context window methods better capture word analogy.

The matrix factorization method is essentially a low-rank approximation of large matrices that records statistical information about a corpus, such as the word-word co-occurrence matrix in which the rows and columns correspond to distinct words and matrix entries record the number of times a word appears in the context of another. The main intuition of GloVe is the observation that ratios of word-word co-occurrence probabilities encode some form of word meaning or similarity. For example, the singular vectors of a word-word co-occurrence matrix can provide an effective low-dimensional vector representation of words that captures word similarity (similar concept to that of PCA). However, a fully-developed word-word co-occurrence matrix can be so large that even the most efficient low-rank decomposition algorithm (e.g. truncated SVD) becomes infeasible to compute.

The central idea of GloVe is to find a word embedding that captures the relationship between pairs of words reflected by the word-word co-occurrence matrix of the corpus. The objective is to minimize the square difference between the dot product of two word vectors with the logarithm of the count of their co-occurrences (and some word-dependent bias terms). In other words, the method assumes that the occurrence counts of two words (or more accurately the ratios of word-word co-occurrence probabilities) is a good representation of the relationship between the two words. Pennington et al. (2014) were able to demonstrate that GloVe can achieve a better performance on word similarity and analogy tasks in most cases compared with Word2Vec and a few other popular methods (e.g. SVD). The authors also provided a theoretical proof that GloVe is related to Word2Vec under certain assumptions and approximations.

Conclusion

Word2Vec as a word embedding technique gained its popularity due to its simplicity in formulation (predicting conext/center words), efficiency in training (shallow learning) and effectiveness in capturing word relationships. FastText extends the idea of Word2Vec and captures subword information using character n-grams. GloVe derives global word representation as vectors from statistics of a whole corpus, and can be shown to be related to the Word2Vec model under certain assumptions. They are all popular and effective methods for representing words using low-dimensional, dense vectors. They are great choices for word embedding in the NLP pipeline, and their usage depends on the specific application.

References

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135-146.

Gutmann, M. U., & Hyvarinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. The journal of machine learning research, 13(1), 307-361.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26, 3111-3119.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1532-1543.