FastText and Word2Vec

Word2Vec

Word2vec understands and vectorizes the meaning of words in a document based on the hypothesis that words with similar meanings in a given context exhibit close distances. CBOW and Skip-gram, learning algorithms of word2vec, both the learning algorithms exhibit Input, Projection, and Output layers although their output derivation processes are different. CBOW predicts the output word from other near word vectors. The basic principle of CBOW involves predicting when a certain word appears via analyzing neighboring words. The projection layer of CBOW projects all words at the same position, and thus, the vectors of all words maintain an average and share the positions of all words. The structure of CBOW exhibits the advantage of uniformly organizing the information distributed in the data set. Skip-gram exhibits a structure for predicting vectors of other words from one word. The basic principle of Skip-gram involves predicting other words that appear around a certain word. The projection layer of the Skip-gram predicts neighboring words around the word inserted into the input layer. The structure of the Skip-gram exhibits the advantage of vectorizing when new words appear. [1]

FastText

FastText embeddings exploit subword information to construct word embeddings. Representations are learnt of character -grams, and words represented as the sum of the- gram vectors. This extends the word2vec type models with subword information. This helps the embeddings understand suffixes and prefixes. Once a word is represented using character -grams, a skipgram model is trained to learn the embeddings.

The key difference between FastText and Word2Vec is the use of n-grams.

References

[1] Word2vec convolutional neural networks for classification of news articles and tweets Beakcheol JangID, Inhwan Kim, Jong Wook Kim*