Dimensionality of Word Embeddings

Last updated: March 18, 2024

Written by: Zhaozhen Xu (https://www.baeldung.com/cs/author/zhaozhenxu)

Reviewed by: Michal Aibin (https://w author)

1. Introduction

Word embedding (/cs/word-embeddings-cbow-vs-skip-gram) is an essential tool for natural language processing (NLP). It has become increasingly popular in recent years due to its ability to capture the semantic meaning of words.

A word embedding is a vector representation of a word in a high-dimensional space. The dimensionality of word embedding is a crucial factor in determining the quality and effectiveness of the embedding.

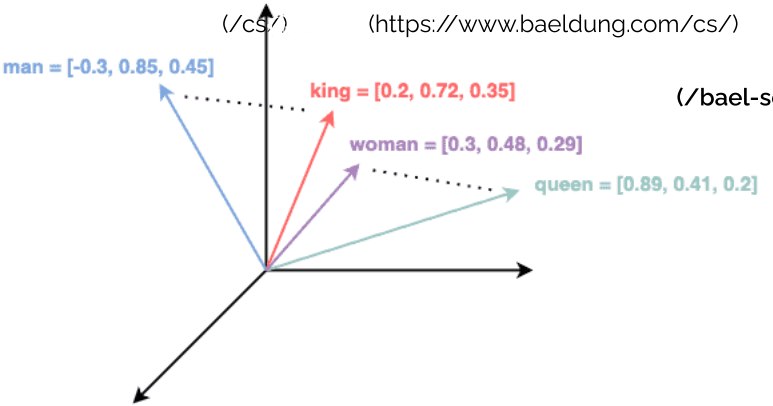
In this tutorial, we'll explain the concept of the dimensionality of a word embedding. We'll also learn how to decide the dimensionality of a word embedding when applying it in the NLP tasks.

2. What Is the Dimensionality of Word Embedding?

In general, **the dimensionality of word embedding refers to the number of dimensions in which the vector representation of a word is defined.** This is typically a fixed value determined while creating the word embedding. **The dimensionality of the word embedding represents the total number of features that are encoded in the vector representation.**

Different methods to generate word embeddings can result in different dimensionality. Most commonly, word embeddings have dimensions ranging from 50 to 300, although higher or lower dimensions are also possible.

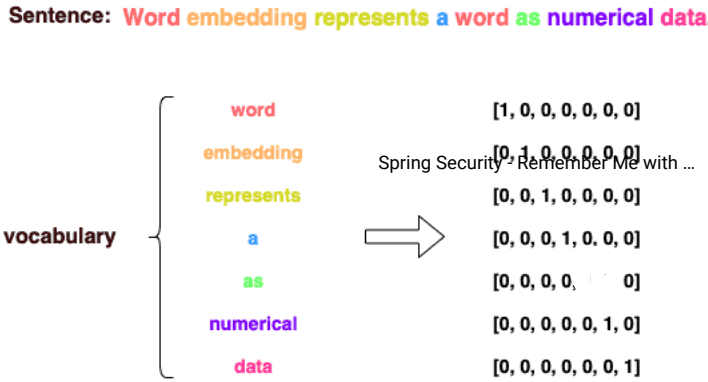
For example, the figure below shows the word embeddings for "king", "queen", "man", and "women" in a 3-dimensional space.



3. Do These Features Mean Anything?

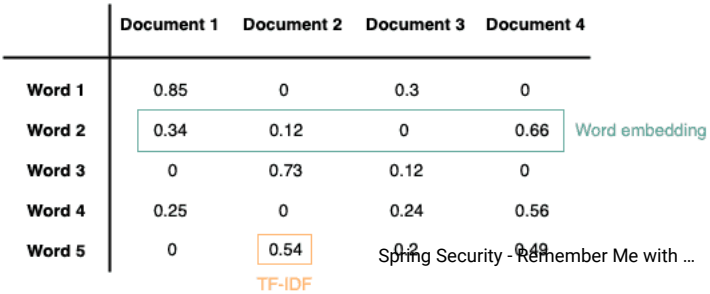
There are multiple ways to generate word representation. **One of the classic methods is one-hot encoding, which represents each word in a vocabulary as a binary vector.** The dimensionality of the embedding is equal to the size of the vocabulary, and each element of the vector corresponds to a word in the vocabulary.

For example, in the sentence "Word embedding represents a word as numerical data.", there are 7 unique words. Thus, the dimensionality of the word embedding is 7:



TF-IDF (/cs/text-sequence-to-vector#4-tf-idf-score-strategy) is another commonly used technique to represent words in natural language processing tasks. It is a numerical statistic that reflects how important a word is to a document in a corpus. While using TF-IDF to generate word representations, we build a TF-IDF matrix that calculates the TF-IDF score for all the vocabulary in the corpus. The size of a TF-IDF matrix depends on the number of documents in the corpus and the size of the vocabulary of unique words.

The figure below gives an example of the word embeddings for "word 2" Given a corpus with 4 documents and 5 unique words, the dimensionality of its word embedding is 4:



	Document 1	Document 2	Document 3	Document 4	
Word 1	0.85	0	0.3	0	
Word 2	0.34	0.12	0	0.66	Word embedding
Word 3	0	0.73	0.12	0	
Word 4	0.25	0	0.24	0.56	
Word 5	0	0.54	0.2	0.49	

TF-IDF

Recently, distributed representations have been widely used in NLP tasks. A distributed representation of a word is a technique for representing the word as a high-dimensional vector in a continuous vector space. The basic idea behind distributed representation is that words with similar meanings or occurring in similar contexts are closer to each other in the high-dimensional space.

With the development of neural networks, scientists introduced more advanced methods that generate distributed word representations, such as Word2vec (/cs/word2vec-word-embeddings#word2vec), GloVe, BERT

(/cs/transformer-text-embeddings#bert), etc.

By using deep learning techniques, the embeddings are learned by the hidden layer of the embedding model. And they are able to capture the semantic and syntactic relationships between words. However, the features of the embedding no long have any meanings. In the meanwhile, there is no universal number for the word embedding dimension. The dimensionality of word embedding depends on different methods and use cases.

4. How Do We Decide the Word Embedding Dimension?

Deciding on the appropriate dimensionality for a word embedding depends on several factors, such as the size and nature of the dataset we are using, the specific NLP task we are working on, and the computational resources available to us.

Here are some general guidelines to consider when deciding on the dimensionality of word embedding.

4.1. Size of the Dataset

Larger datasets can support higher-dimensional embeddings as they provide more training data to inform the model. As a rule of thumb, a dataset with less than 100,000 sentences may benefit from a lower-dimensional embedding (e.g., 50-100 dimensions), while a larger dataset may benefit from a higher-dimensional embedding (e.g., 200-300 dimensions).

4.2. NLP Task

The NLP task we are working on can also inform the dimensionality of our word embedding.

For example, tasks that require a high level of semantic accuracy, such as sentiment analysis or machine translation, may benefit from a higher-dimensional embedding. However, easier tasks such as named entity recognition or part-of-speech tagging may not require as high of a dimensional embedding.

4.3. Computational Resources

The dimensionality of our word embedding will also affect the computational resources required to train and use the model. Higher-dimensional embeddings require more memory and processing power, so it's important to consider the computational resources available when deciding on the dimensionality of our word embedding.

4.4. Find the Right Number of Dimensions

Ultimately, the best approach is to experiment with different dimensionalities and evaluate the performance of our model on a validation set. We can gradually increase or decrease the dimensionality until we find the optimal balance between semantic accuracy and computational efficiency for our specific use case.

5. Summary

In this article, we talked about word embedding and the dimensionality of word embedding.

The dimension of the word embedding produced by classic word embedding methods, such as one-hot encoding and TF-IDF, is highly dependent on the size of the corpus and vocabulary. On the other hand, for distributed representations such as Word2vec and GloVe, the optimal dimensionality depends on factors such as the size of the training dataset, the computational resources available, and the specific task at hand. And it is usually set empirically.

(/cs/) (https://www.baeldung.com/cs/)

(/bael-search)

Spring Security - Remember Me with ...

CATEGORIES

ALGORITHMS (/CS/CATEGORY/ALGORITHMS)
ARTIFICIAL INTELLIGENCE (/CS/CATEGORY/AI)
CORE CONCEPTS (/CS/CATEGORY/CORE-CONCEPTS)
DATA STRUCTURES (/CS/CATEGORY/DATA-STRUCTURES)
LATEX (/CS/CATEGORY/LATEX)
NETWORKING (/CS/CATEGORY/NETWORKING)
SECURITY (/CS/CATEGORY/SECURITY)

SERIES

GRAPHS TUTORIAL (HTTPS://WWW.BAELDUNG.COM/CS/GRAPHS-SERIES)
NEURAL NETWORKS SERIES (HTTPS://WWW.BAELDUNG.COM/CS/NEURAL-NETWORKS-SERIES)
LATEX SERIES (HTTPS://WWW.BAELDUNG.COM/CS/LATEX-SERIES)

(/cs/) (https://www.baeldung.com/cs/)

(/bael-search)

ABOUT

ABOUT BAELDUNG (HTTPS://WWW.BAELDUNG.COM/ABOUT)
THE FULL ARCHIVE (/CS/FULL_ARCHIVE)
EDITORS (HTTPS://WWW.BAELDUNG.COM/EDITORS)
OUR PARTNERS (HTTPS://WWW.BAELDUNG.COM/PARTNERS/)
PARTNER WITH BAELDUNG (HTTPS://WWW.BAELDUNG.COM/PARTNERS/WORK-WITH-US)
EBOOKS (HTTPS://WWW.BAELDUNG.COM/LIBRARY/)
FAQ (HTTPS://WWW.BAELDUNG.COM/LIBRARY/FAQ)
BAELDUNG PRO (/MEMBERS/)

TERMS OF SERVICE (HTTPS://WWW.BAELDUNG.COM/TERMS-OF-SERVICE)
PRIVACY POLICY (HTTPS://WWW.BAELDUNG.COM/PRIVACY-POLICY)
COMPANY INFO (HTTPS://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO)
CONTACT (/CONTACT)