Word2Vec

To process natural language text and extract usable information from the provided word, a phrase must be turned into a set of real numbers (a vector) using machine learning and deep learning algorithms.

When it comes to finding word predictions, word similarities, and semantics, Word Embeddings and Word vectorization are two methods used in Natural Language Processing (NLP).

When words are turned into numerical values, it is called Vectorization.Word embeddings are beneficial in the following situations.

- Compute similar words
- Text classifications
- Document clustering/grouping
- Feature extraction for text classifications
- Natural language processing.

Once the words have been transformed into vectors, we can use techniques like Euclidean distance and Cosine Similarity to see which ones are related. As a next step, let's explore the various methods for turning phrases into vectors.

Word embeddings derived from pre-trained algorithms such as,

- Word2Vec — From Google
- Fasttext — From Facebook
- Glove — From Stanford

Word2Vec

Word2Vec — Tomas Mikolov and a Google research team developed this approach in 2013 using word representations in vector space.

Word2Vec was developed for the following reasons:

Words are typically treated as atomic units in NLP systems. Existing systems have a flaw in that there is no concept of word similarity. It also outperforms on smaller datasets, such as those with less than a few billion records, because the method is optimised for tiny, simpler datasets.

New techniques employ a neural network architecture to train complicated data models with a larger dataset and outperforms huge datasets with billions of words and millions of words of vocabulary. They use modern techniques.

It's a useful method for gauging how well the generated vector representations are performing. This works with words that have numerous degrees of similarity that tend to close with each other.

Syntactic Regularities: This term refers to the rectification of grammatical sentences.
Semantic Regularities: This term refers to the meaning of the vocabulary symbols that are arranged in that structure.

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

To test out the proposed method, it was discovered that word representation similarity goes well beyond grammatical structure and performs admirably for algebraic operations on word vectors. Take, for instance,
Vector(“King”) — Vector(“Man”)+Vector(“Woman”) = Word(“Queen”)
where “Queen” is the closest result vector of word representations.

The accuracy and processing complexity of the following word representation model architectures are maximised while remaining minimal. The models are,

- FeedForward Neural Net Language Model (NNLM)
- Recurrent Neural Net Language Model (RNNLM)

All the above-mentioned models are trained using Stochastic gradient descent and backpropagation.