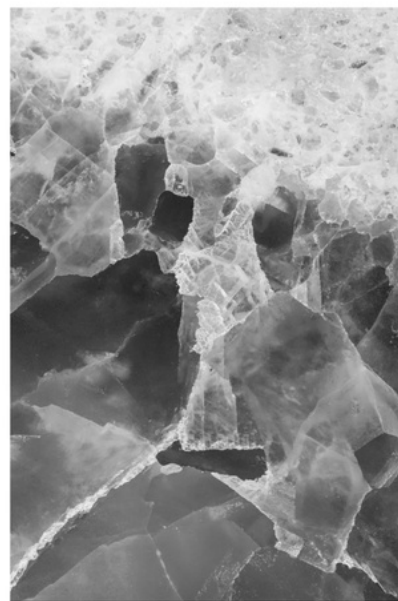
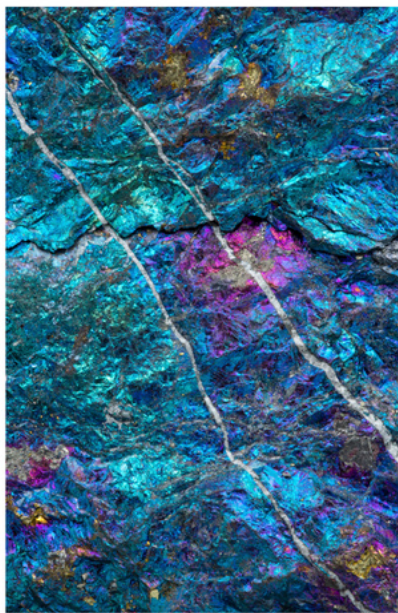


Obsidian

Twitter Instagram LinkedIn Share



Copyright © All rights reserved.

Using Format

# Word Embeddings in NLP and its Applications

Taken from: <https://www.kdnuggets.com/2019/02/word-embeddings-nlp-applications.html>

# Word Embeddings in NLP and its Applications



Word embeddings such as Word2Vec is a key AI method that bridges the human understanding of language to that of a machine and is essential to solving many NLP problems. Here we discuss applications of Word2Vec to Survey responses, comment analysis, recommendation engines, and more.

Word embeddings are basically a form of word representation that bridges the human understanding of language to that of a machine. Word embeddings are distributed representations of text in an n-dimensional space. These are essential for solving most NLP problems.

Domain adaptation is a technique that allows Machine learning and Transfer Learning models to map niche datasets that are all written in the same language but are still linguistically different. For example, legal documents, customer survey responses, and news articles are all unique datasets that need to be analyzed differently. One of the tasks of the common spam filtering problem involves adopting a model from one user (the source distribution) to a new one who receives significantly different emails (the target distribution).

The importance of word embeddings in the field of deep learning becomes evident by looking at the number of researches in the field. One such research in the field of word embeddings conducted by Google led to the development of a group of related algorithms commonly referred to as Word2Vec.



Word2Vec one of the most used forms of word embedding is described by Wikipedia as:

“Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.”

In this post, we look at some of the practical uses of Word Embeddings (Word2Vec) and Domain Adaptation. We also look at the technical aspects of Word2Vec to get a better understanding.



## Analyzing Survey Responses

Word2Vec can be used to get actionable metrics from thousands of customers reviews. Businesses don't have enough time and tools to analyze survey responses and act on them thereon. This leads to loss of ROI and brand value.

Word embeddings prove invaluable in such cases. Vector representation of words trained on (or adapted to) survey data-sets can help embed complex relationship between the responses being reviewed and the specific context within which the response was made. Machine learning algorithms can leverage this information to identify actionable insights for your business/product.

Check out SmartReader, a simple Excel-based tool by ParallelDots that automates Survey Response Analysis and can be used by anyone.



## Analyzing Verbatim Comments

Machine learning with the help of word embeddings has made great headway in the domain of analysis of verbatim comments. Such analyses are very important for customer-centric enterprises.

When you're analyzing text data, an important use case is analyzing verbatim comments. In such cases, the data scientist is tasked with creating an algorithm that can mine customers' comment or review.



Word embeddings like Word2Vec are essential for such Machine Learning tasks. Vector representations of words trained on customer comments and reviews can help map out the complex relations between the different verbatim comments and reviews being analyzed. Word embeddings like Word2Vec also help in figuring out the specific context in which a particular comment was made. Such algorithms prove very valuable in understanding the buyer or customer sentiment towards a particular business or social forum.

Check out the SmartReader by ParallelDots to make headway in the process of automating your enterprise's Verbatim comments analysis.



# Music/Video Recommendation System



The way we experience content has been revolutionized by streaming services available across the Internet. In the past, recommendations focused on presenting you with content for future use. The modern streaming platforms instead focus on recommending content that can and will be enjoyed at the moment. The streaming models bring to the table new methods of discovery in the form of personalized radio and recommended playlists.

The focus here is on generating sequences of songs that gel. To enhance user-experience the recommendation system's model should capture not only what songs similar people generally interested in, but also what songs are listened to frequently together in very similar contexts.

Such models make use of Word2Vec. The algorithm interprets a user's listening queue as a sentence with each song considered as a word in the sentence. When a Word2Vec model is trained on a such a dataset what we mean is that each song that the user has listened to in the past and the song one is listening to at present somehow belong to the same context. Word2Vec accurately represents each song with a vector of coordinates that maps the context in which the song or the video is played.

For those of you who want to delve into the technical aspects of how Word2Vec works, here is what ParallelDots's in-house experts view the subject.



## Technical Aspect of Word Embeddings

A common practice in NLP is the use of pre-trained vector representations of words, also known as embeddings, for all sorts of down-stream tasks. Intuitively, these word embeddings represent implicit relationships between words that are useful when training on data that can benefit from contextual information.

Consider the example of the Word2Vec skip-gram model by Mikolov et al. — one of the two most popular methods of training word embeddings (the other being GloVe). The authors pose an analogical reasoning problem which essentially requires asking the question: “Germany is to Berlin as France is to \_\_\_?”. When you consider each of these words as a vector, the answer to the given problem is simply given by the formula

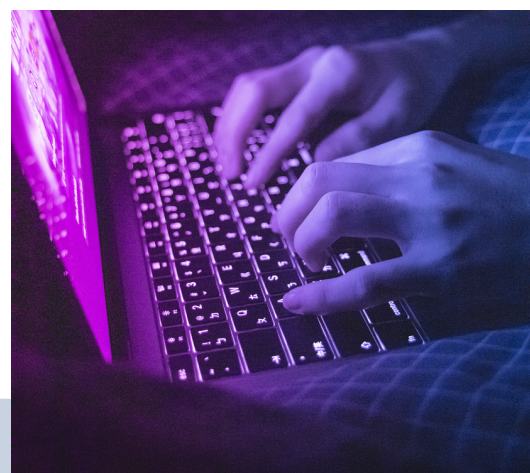


$$\text{vec}(\text{"Berlin"}) - \text{vec}(\text{"Germany"}) = x - \text{vec}(\text{"France"})$$

That is, the distance between the sets of vectors must be equal. Therefore,

$$x = \text{vec}(\text{"Berlin"}) - \text{vec}(\text{"Germany"}) + \text{vec}(\text{"France"})$$

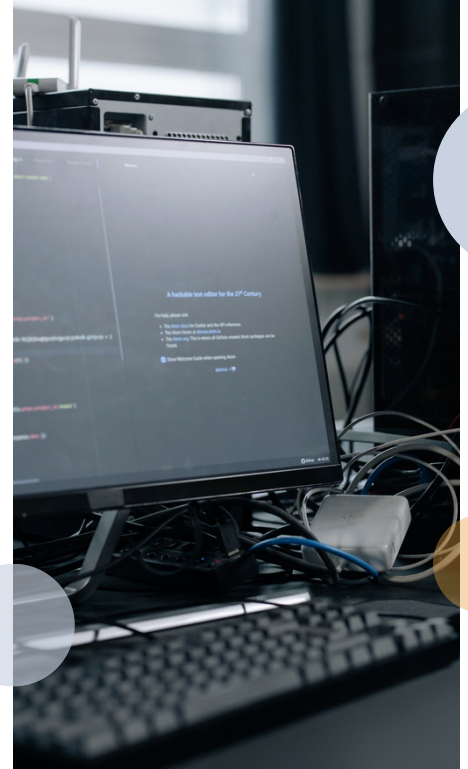
Given that the vector representations are learned correctly, the required word is given by the vector closest to the point obtained. Another implication of this is that words with similar semantic and/or syntactic meanings will group together.



## RETROFITTING

While general-purpose datasets often benefit from the use of these pre-trained word embeddings, the representations may not always transfer well to specialized domains. This is because the embeddings have been trained on massive text corpus created from Wikipedia and similar sources.

For example, the word python means something else in the everyday context, but it means something else entirely in the context of computer programming. These differences become even more relevant when you are building models to analyze context critical data such as in medical and legal notes.



One solution is to simply train the GloVe or skip-gram models on the domain-specific datasets, but in many cases sufficiently large datasets are not readily available to obtain practically relevant/meaningful representations.

The goal of retrofitting is to take readily available pre-trained word vectors and adapt them to your new domain data. The resulting representations of words are arguably more context-aware than the pre-trained word embeddings.