Published in Towards Data Science

Jon Gimpel Follow

Jul 26, 2022 · 7 min read · Listen

A PRIMER ON WORD EMBEDDINGS

What's Behind Word2vec

An outline of ideas and equations for word embeddings

Photo by Stefan Steinbauer on Unsplash

The field of Natural Language Processing (NLP) requires knowledge of linguistics, statistics, and computer science. As such, it can be challenging to begin a new study or project without significant background research in the disciplines that you're less familiar with. It can also be challenging to move seamlessly across these disciplines because their nomenclature and equation formats differ.

I came to NLP from statistics wanting to learn how words in a collection of texts (*corpus*) get transformed into usable data in the form of vectors (*word em* following two examples:

```
aardvark [  0.7660651 -0.9571466 -0.4889298  ...  -0.1600012 ]
zulu     [ -0.4566793  0.7392789  0.5158788  ...   0.0398366 ]
```

I wanted to understand what's really being measured, what's most important, and what gets compromised in the transformation to vector data generated by the algorithms in Word2vec, fastText, and modern contextualized word representations.

Fundamentally, *what do the numbers in word embeddings values really represent*?

In addition, many words have multiple meanings (*senses*), so I was especially interested in learning how a word with multiple senses and only one vector, of say 100 dimensions, could still be statistically valid. I then wanted to be able to program my own word embedding training algorithm based on the Word2vec model, so I could explore sense-specific representations.

This article is the first of a series of articles. It introduces the concepts of how words relate to each other through their proximity in text and the theory behind creating data about word relationships. In this series, I'll also translate NLP concepts from linguistics and computer science into a statistical perspective.

Much of today's developments in NLP center on deep learning artificial intelligence algorithms, but anyone entering the field should have a clear conceptual grasp of all the building blocks.

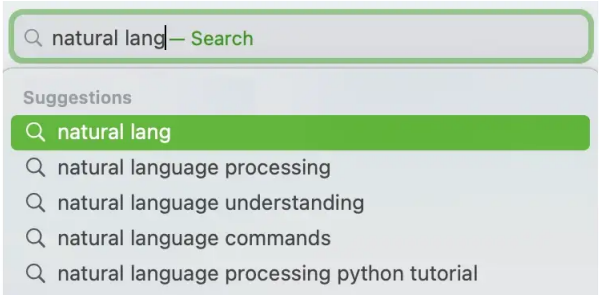
The articles in this series are:

1. **What's Behind Word2vec** (*this article*)
   An outline of ideas and equations for word embeddings *(7 min read)*
2. **Words into Vectors**
   Concepts for word embeddings *(13 min read)*
3. **Statistical Learning Theory**
   The basis for neural networks *(14 min read)*
4. **The Word2vec Classifier**
   How word embeddings are trained *(15 min read)*
5. **The Word2vec Hyperparameters**
   A set of creative reweightings *(6 min read)*
6. **Characteristics of Word Embeddings**
   And the problem of antonyms *(11 min read)*

Before we delve into NLP and word embeddings, let's briefly look at the landscape leading up to the creation of Word2vec. We'll focus on Word2vec because it popularized the type of word embeddings in use today.

Word Proximity as the Basis for Word Definitions

One of the basic applications of a computational language model is predicting words in a sentence, as for example, in the auto-complete feature of search engines and messaging apps:



(image by author)

Such models can be language models that probabilistically predict each word's appearance by tabulating all word sequences in a large corpus of text, but from an implementation perspective, processing and storing all this information is impractical due to the volume of data. For example, the size of even a simple collection of data, such as a *co-occurrence matrix* of how often pairs of words appear together in each document in a set of documents, would be the square of the total number of unique words, perhaps hundreds of thousands of words squared.

Word embeddings, which are representations of words using vectors, help reduce these computational challenges. Instead of storing all the information about all the words in all documents, word embeddings leverage creative data processing and statistical dimension reduction techniques to approximate the relationships of the words.

A fascinating property of these modern, machine-learned word embeddings is that, when they are applied to language models, they predict not just word sequences based on proximity frequency, but in a way, word meanings.

Word embeddings are a manifestation of philosopher Ludwig Wittgenstein's idea that "the meaning of a word is its use in the language" (Wittgenstein, 1953). In 1957, the linguist John Rupert Firth put this concept more concretely as:

> "You shall know a word by the company it keeps."

John Rupert Firth (1890–1960)
(Photo from article by Scott, N. (1961). Bulletin of the School of Oriental and African Studies, 24(3), 412–418, reproduced with permission)

So, a word can be defined via the words with which it typically appears. For example, because the word 'rock' might appear, depending on the context, alongside words such as 'earth' and 'music', both earth and music have something to do with the definition of rock. Today, this concept is known as the *distributional hypothesis* in linguistics (Perone, 2018).

> *"But from a Statistical NLP perspective, it is more natural to think of meaning as residing in the distribution of contexts over which words and utterances are used. … Under this conception, much of Statistical NLP research directly tackles questions of meaning."* (Manning and Schütze, 1999)

Similarly to how a dictionary defines all words merely by their relationship to each other, a word embedding matrix uses numerical values to define its words by their proximity in use.

Why Data Models Began to Supercede Rules-based Models

The field of Natural Language Processing (NLP) aims to have computers interact using human language. Many approaches have been taken to implement human language in computers, and ideas in linguistics have evolved in their reliance on sophisticated computer algorithms. Language models that are strongly founded in language structure, rules, and logic are often too processing intensive or complex to be practical, and shortcuts that work well computationally often have obvious linguistic weaknesses.

Statistics plays a prominent role in empirical NLP, not just in the analysis of written and oral language data, but in the statistical learning theory behind the machine learning that is increasingly applied to analyze large corpora (Stewart, 2019). Still, the value of statistical NLP was prominent even before the acceleration of machine learning and artificial intelligence (AI).

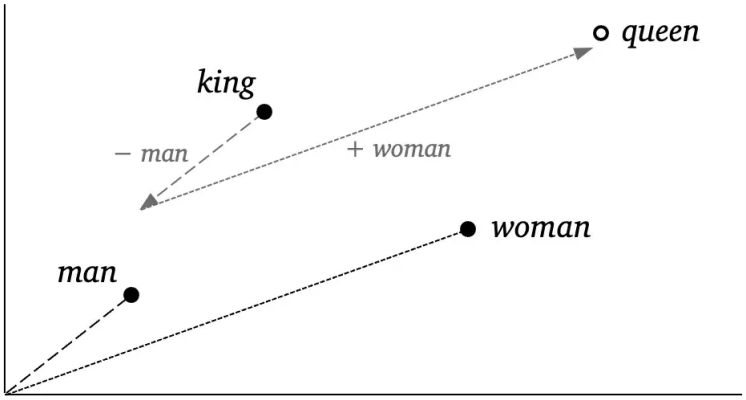
> *"Statistical [NLP] models are robust, generalize well, and behave gracefully in the presence of errors and new data."* (Manning and Schütze, 1999)

The Debut of Word2vec: What Made it so Transformative

In 2013, a leap in NLP enthusiasm arrived with the publication of two papers by Mikolov et al. at Google introducing Word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b). Word2vec uses a shallow neural network to produce word embeddings that perform especially well with the added benefit of a huge increase in computing efficiency. With Word2vec, a set of word vectors can be created from a relatively large corpus in any language with just a personal computer. Another prominent feature of Word2vec is the observation that the word vectors cluster synonyms and related words nearby in the vector space. Plus, the vectors appear to have mathematical properties. For example, by adding the vector values, one finds the following famous equation in computational linguistics:

$king - man + woman \approx queen$

In two dimensions, this equation might look like the following:



Word Vector Mathematics Concept
(image by author)

These Word2vec vectors improved many applications of NLP tasks, and a plethora of research ensued to study the properties and implications of neural network word embeddings. The ideas generated through this research eventually led to more powerful AI models with context-sensitive embeddings (such as AllenNLP ELMo, OpenAI's GPT, and Google's BERT).

Summary

In this article, we learned the linguistics theory that word proximity in use is related to word meaning and that rules for natural language can be impractical to implement. We also learned that using word proximity to create word vectors can yield a manageable dataset with useful properties.

In the next article, **Words into Vectors**, we'll review the foundational concepts behind the creation of word embeddings.

This article was the 1st in the series **A primer on word embeddings:**
1. What's Behind Word2vec | 2. Words into Vectors |
3. Statistical Learning Theory | 4. The Word2vec Classifier |
5. The Word2vec Hyperparameters | 6. Characteristics of Word Embeddings

More on this Topic: For each article in this series, I'll recommend a key reference for additional information on the topic. For this article, you might especially enjoy: Perone, C. S. (2018). NLP Word Representations and the Wittgenstein Philosophy of Language. *Terra Incognita*.

References

Firth, J. R. (1957). A synopsis of linguistic theory, 1930–1955. In Firth (ed), *Studies in Linguistic Analysis*, Special Volume of the Philological Society, pages 1–32. Oxford, England: Basil Blackwell Publishing.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press.

Mikolov, T., Corrado, G, Chen, K., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. Available at arXiv:1301:3781v3.

Mikolov, T., Corrado, G, Chen, K., Sutskever, I., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. Available at arXiv:1310.4546v1.

Perone, C. S. (2018). NLP Word Representations and the Wittgenstein Philosophy of Language. *Terra Incognita*.

Stewart, M. (2019). The Actual Difference Between Statistics and Machine Learning. *Towards Data Science*.

Wittgenstein, L. (1953). *Philosophical Investigations*. Oxford, England: Basil Blackwell Publishing.

*Figures and images are by the author, unless otherwise noted.

NLP Word 2 Vec Machine Learning Word Embeddings Primer Editors Pick

Thanks to Katherine Prairie and Ben Huberman