# Distributional Feature Representations

Jonathan May

September 6, 2022(Prepared for Fall 2022)

**Discussion Question: What important-seeming features of words are not well captured by the distributional hypothesis?**

This material was pulled from a longer lecture on semantics, so when we get to semantics there may be less to say!

## 1 Where did those features come from?

In the last notes we started with $\mathbf{f}$ but nothing much was said about where $\mathbf{f}$ came from. The answer is that they can be learned just like the other parameters, I just skipped over that part before. In a feed-forward network, where the input size is fixed, we can consider the input to be some sequence of $k$ words.[1] Let $kd = f$; we call $d$ the feature (or embedding) size. Consider the vocabulary to be $W$ and assume that every word $w$ is assigned an index $i$ from 0 to $|W| - 1$. We can represent $w$ as a *one-hot vector* $[0, 0, \ldots, 0, 1, 0, 0, \ldots, 0]$ that is $|W|$ long, with all units set to 0 except unit $i$, which is 1. Let $E$ be a $|W| \times d$ matrix. Then to form $f$ we take each one-hot vector for each word in the input sequence and multiply by $E$; we concatenate the sequence of resulting vectors together. Note that in practice you don't implement this way – multiplying by a one-hot is equivalent to indexing from an array...it's just a lot more space-inefficient.

Ok, but why are we doing this? The features could have been the sparse words (or a bag of words) but intuitively it would be nice if we had a fixed set of features for each word, such that words that are similar are close to each other (where 'close' means 'cosine of the angle between their vectors is small'). Otherwise 'cat' and 'feline' are as different from each other as 'cat' and 'rutabaga.' Enter the Distributional Hypothesis!

### 1.1 Distributional Methods

A totally different way to understand word similarity is based on a famous quote by linguist John Rupert Firth (1957): "You shall know a word by the company it keeps." That is, words are similar if the words they are near are similar.

Intuition from Zelig Harris (another linguist) in 1954: "oculist and eye-doctor occur in almost the same environments...thus we say they are synonyms."

Here's another example:

[1] I'm not being careful about variables now relative to last notes; we used up all the letters.

```
A bottle of tesgüino is on the table
Everybody likes tesgüino
Tesgüino makes you drunk
We make tesgüino out of corn.
```

What do you think tesgüino is?

## 1.2 Word co-occurrence matrices and mutual information

Let's try this first comparing documents and words:

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 1 | 8 | 15 |
| soldier | 2 | 2 | 12 | 36 |
| fool | 37 | 58 | 1 | 5 |
| clown | 6 | 117 | 0 | 0 |

Notice the usage patterns of 'fool' and 'clown' vs 'battle' and 'soldier.' Notice also the similarity of some of the plays.

Important difference from thesaurus-based approaches now; we're losing the ability to distinguish between senses of the same word form (there are ways to try to get these back but won't cover them here and the 'hard decision' approach doesn't work that well...but we will revisit this when we discuss contextualized representations).

We can also make such a table for smaller contexts, such as a four-word window.

| | aardvark | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|---|
| apricot | 0 | 0 | 0 | 1 | 0 | 1 |
| pineapple | 0 | 0 | 0 | 1 | 0 | 1 |
| digital | 0 | 2 | 1 | 0 | 1 | 0 |
| information | 0 | 1 | 6 | 0 | 4 | 0 |

In this table the number of times a word in the column was seen within four words of the word in the row is listed in the cell.

In reality this table is $|V| \times |V|$ but the vast majority of cells are 0 (very sparse). Notice again how similar words have similar vector patterns.

This is so because of two co-occurrence phenomena:

- Syntagmatic (first-order) association (surface similarity): sets of words all occur near each other, somewhat interchangeably. E.g. 'wrote', 'book', and 'poem' all tend to occur near each other so they are likely to have similar patterns (example: "Whether a book or a poem, what Jane Austen wrote will live for generations.").

- Paradigmatic (second-order) association (paradigm similarity): words don't necessarily occur near each other but nevertheless do have similar neighbors. E.g. 'wrote', 'said', 'remarked' all share a 'paradigm' of words they occur near. (example: "The candidate remarked that the troops were important." "The candidate said he valued the importance of the troops." "The candidate wrote that the troops mattered a lot to him.")

It has been observed that a narrow co-occurrence window (1-3) will tend to give words with similar *syntactic* properties more similar vectors and with a wider window (4-10) more *semantic* and not necessarily syntactic similarity. Think 'orange/apple/lemon/carrot' for the former and 'kill/death/killing/killed' for the latter. These are not hard and fast rules.

Not all co-occurring words are equally informative! Consider 'the' and 'of' which occur many times very frequently with other words. It's better to ask which words are particularly informative. Specifically, if words occur more frequently than they do by chance, this is interesting to us[2]. We specifically define *pointwise mutual information* for words $w_1, w_2$:

$$MI(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \tag{1}$$

If $w_1$ and $w_2$ are IID, we'd expect $P(w_1, w_2) = P(w_1)P(w_2)$. If this is so, then $MI = 0$. If the words co-occur more likely than expected, i.e. $P(w_1, w_2) > P(w_1)P(w_2)$, then $MI > 0$. If they occur less frequently $MI < 0$. This last element is often ignored; we don't really know what it means to be some degree of 'unrelated' plus the resolution needed to detect events *less* likely than the product of two events necessitates very large corpora. Typically we instead study *positive pointwise mutual information*:

$$PPMI(w_1, w_2) = \max(\log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)}, 0)$$

Here's a worked example. Using the table above of frequency counts $f_{ij}$ for word $i$ in context of word $j$, we can calculate joint probability, word probability, and context probability, as:

$$p_{\text{joint}}(i, j) = \frac{f_{ij}}{\sum_w \sum_c f_{wc}}$$
$$p_{\text{word}}(i) = \frac{\sum_c f_{ic}}{\sum_w \sum_c f_{wc}}$$
$$p_{\text{context}}(j) = \frac{\sum_w f_{wj}}{\sum_w \sum_c f_{wc}}$$

| | $p(w,c)$ | | | | | $p(w)$ |
|---|---|---|---|---|---|---|
| | computer | data | pinch | result | sugar | |
| apricot | 0 | 0 | .05 | 0 | .05 | .11 |
| pineapple | 0 | 0 | .05 | 0 | .05 | .11 |
| digital | .11 | .05 | 0 | .05 | .05 | .21 |
| information | .05 | .32 | 0 | .21 | 0 | .58 |
| $p(c)$ | .16 | .37 | .11 | .26 | .11 | |

PMI(information, data) = $\log \frac{.32}{.37 \times .58} = .57$

Here are all PMIs:

[2] see also TF*IDF, another way to formulate the same idea

| | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|
| apricot | 0 | 0 | 2.25 | 0 | 2.25 |
| pineapple | 0 | 0 | 2.25 | 0 | 2.25 |
| digital | 1.66 | -.56 | 0 | -.07 | 0 |
| information | -.8 | .57 | 0 | .47 | 0 |

To get PPMI, replace the negative values with 0. The unfilled boxes are also

## 1.3 Cosine similarity

A nice number for characterizing the closeness of two vectors is the *cosine* of these vectors. Each word is represented as a vector in $|V|$-space. If the angle they make is small, the cosine is close to 1. Cosine is just a normalized dot-product. Simple dot product isn't a great way to calculate closeness, because longer vectors (i.e. with high values in some dimensions) will lead to larger dot product. Cosine normalizes this:

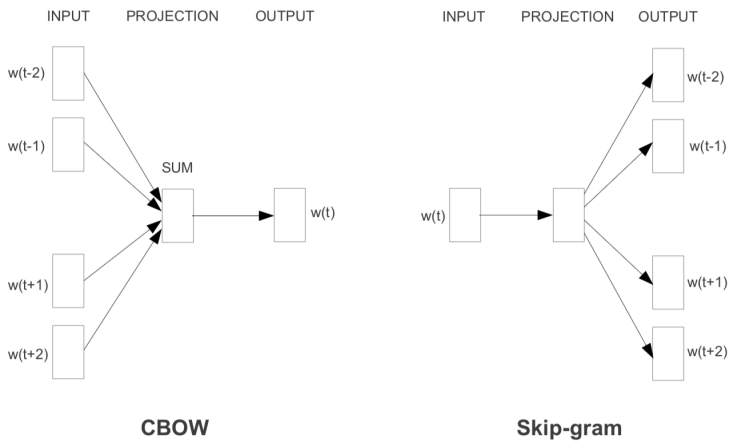
$$\cos(a, b) = \frac{a \cdot b}{|a||b|}$$

where

$$|x| = \sqrt{\sum_i x_i^2}$$

## 1.4 Neural(-inspired) distributional representations

An issue with PPMI is the vectors are very sparse and the dimensions very large. We previously saw embeddings were lower-dimensional dense representations of words. We want embeddings for words to be aware of the contexts in which these words occur. One way to do this is inspired by feed-forward neural networks. Mikolov's *skip-gram* is like a miniature version of the FFNNLM. It contains a word embedding matrix $E$ and an output matrix $O$ but no hidden matrix and no nonlinear function. Given a word $w$ and its context word $c$, the logit for $c$ is simply $(E_w O)_c$. Given some text, training data is formed by taking $c$ to be any word within some range $r$ before or after $w$. An alternative framework called the *continuous bag-of-words* sums together the embeddings of context words within $r$ of $w$ to predict it. In other words, for $r = 2$, the logit for $w$ is $((E_{w-2} + E_{w-1} + E_{w+1} + E_{w+2})O)_w$.




.

These models, along with some techniques for training them very quickly, are known collectively as *Word2Vec* (w2v). Some nice properties observed with them is that one could do *vector math*; the vector formed by subtracting *big-biggest* is very similar to that formed from *small-smallest*. To this end, the W2v authors created an analogy test set. To evaluate vectors, you consider an analogy like "brother:sister::grandson:granddaughter." You calculate *grandson+(sister-brother)*. If the closest embedding to that vector is *granddaughter* the relationship has been captured. The relationship types are shown below, as are some results.

| 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 |

| Model Architecture | Semantic-Syntactic Word Relationship test set | |
|---|---|---|
| | Semantic Accuracy [%] | Syntactic Accuracy [%] |
| RNNLM | 9 | 36 |
| NNLM | 23 | 53 |
| CBOW | 24 | 64 |
| Skip-gram | 55 | 59 |

It turns out the product of the w2v embeddings can be shown to be closely related to a PMI table.