

Author: Benjamin Smidt  
Created: September 24, 2022  
Last Updated: September 24, 2022

## Assignment 1: Exploring Word Vectors

Note to the reader. This is my work for assignment one of Stanford's course CS 224N: Natural Language Processing with Deep Learning. You can find the lecture Winter 2021 lectures series on YouTube [here](#). This document is meant to be used as a reference, explanation, and resource for the assignment, not necessarily a comprehensive overview of Word Vectors. If there's a typo or a correction needs to be made, feel free to email me at [benjamin.smidt@utexas.edu](mailto:benjamin.smidt@utexas.edu) so I can fix it. Thank you! I hope you find this document helpful :).

## Contents

<b>1</b>	<b>Count-Based Word Vectors</b>	<b>2</b>
1.1	Distinct Words . . . . .	2
1.2	Compute Co-Occurrence Matrix . . . . .	2
1.3	Reduce to K Dimensions . . . . .	3
1.4	Plot Embeddings . . . . .	3
1.5	Co-Occurrence Plot Analysis . . . . .	3
<b>2</b>	<b>Prediction-Based Word Vectors</b>	<b>3</b>
2.1	GloVe Plot Analysis . . . . .	3
2.2	Words with Multiple Meanings . . . . .	4
2.3	Synonyms and Antonyms . . . . .	4
2.4	Analogies with Word Vectors . . . . .	4
2.5	Finding Analogies . . . . .	5
2.6	Incorrect Analogy . . . . .	5
2.7	Guided Analysis of Bias in Word Vectors . . . . .	5
2.8	Independent Analysis of Bias in Word Vectors . . . . .	5
2.9	Thinking About Bias . . . . .	6
<b>3</b>	<b>Resources</b>	<b>6</b>

# 1 Count-Based Word Vectors

## 1.1 Distinct Words

Our first function is pretty simple. Our goal is to flatten the list of lists of words into one long list. From there we remove all the duplicate words and return both the sorted list and its length. See the notebook for links on using list comprehension (to flatten the list of list of words). Python's *set* data structure comes in very handy for removing duplicate words. Finally, we finish by having python convert that set back to a list for us and using the *sorted* function to sort that list.

## 1.2 Compute Co-Occurrence Matrix

This function is slightly more complicated but honestly if you've done some python programming it's fairly trivial. We first grab our corpus as well as our *words* word list and it's length (*num-words*) with the *distinct-words()* function we just implemented. Next we create a dictionary mapping between each word in *words* with some number *i* and store it in the dictionary *word2ind*. The word's number *i* will serve as its index along both dimensions of our co-occurrence matrix *M*. Then we initialize our co-occurrence matrix *M*, which has dimensions *num-words* x *num-words* with the first dimension being the center word and the second being the context words (it doesn't really matter which one is which, that's just how I'm thinking about it).

We fill our co-occurrence matrix using a for-loop to iteratively compute the number of context words for a given center word. For each center word we move backward one word in the document and use our dictionary *word2ind* to find the proper row (center word) and column (context word) in *M*, and increment *M[center-word-index, context-word-index]* by one. We do this until we've moved backward by *window-size* or until we hit *START*. We repeat the same procedure moving forward except moving forward a word and stopping at *END* or until we've reached a number of words equalling *window-size*. Finally, we return our co-occurrence matrix *M* and our dictionary mapping *word2ind*.

### 1.3 Reduce to K Dimensions

I'll quite honest here, I'm not very familiar with PCA or SVD. For this assignment it's not necessary to know exactly how it works. The point is that we're extracting the most significant data from our co-occurrence matrix to reduce its dimensionality. Regarding code, just use the documentation from sklearn on how to call it and note the description in the notebook about SVD and Truncated SVD options in different libraries.

### 1.4 Plot Embeddings

This function is pretty simple. All we do is create a function that will things onto two dimensions. For our purposes, that will be plotting the two most significant dimensions of  $U$  from our singular value decomposition.

### 1.5 Co-Occurrence Plot Analysis

We don't actually do anything but plot the output for this cell. The interesting part though is the similarity or dissimilarity of words. It makes sense that the countries are clustered close together, that oil and energy synonymous, and (less so) that petroleum and industry are synonymous. However, I'd expect "barrels" and "bpd" (barrels per day) to have a much closer meaning considering barrel is literally in the acronym of "bpd". Additionally, I feel like "bpd" should generally be closer to the other oil related words since "bpd" is an exclusively oil/petroleum word used in the energy industry.

## 2 Prediction-Based Word Vectors

### 2.1 GloVe Plot Analysis

We can see that the GloVe plot produced is somewhat different than that produced by our co-occurrence matrix and SVD. The first thing I noticed was that although the countries are close to each other, Kuwait is much farther from Ecuador and Iraq than in our co-occurrence plot. Furthermore, petroleum, Ecuador, and Iraq are *very* close together. GloVe suggests that petroleum is more synonymous with Iraq and Ecuador than our co-occurrence matrix.

Another difference that stands out is how closely it place the words energy and industry. They appear in the exact same place which wasn't the case at all in our co-occurrence matrix. Finally, we see that "bpd" and "barrels" have roughly the exact same distance between them as our co-occurrence plot. This is quite interesting to me. I'm not totally sure what to think about that other than that the math suggests that's how they should be related.

## 2.2 Words with Multiple Meanings

I discovered distortion which has a variety of meanings. It includes: exaggeration, misrepresentation, amplification, and vibration. Exaggeration and vibration are particularly different but you can see that GloVe found some significantly different meanings in the way distortion is used. Many of the words I tried didn't work because GloVe only learns word meaning based on the dataset or documents used. Thus, if a word is only ever used in a particular context for a given dataset, GloVe will only be able to learn the word's meaning in that particular context.

## 2.3 Synonyms and Antonyms

Counterintuitively, you can see that *timid* and *shy* have a large cosine distance than *timid* and *pushy*. By our framework, this would suggest that *timid* is closer to *pushy* than *shy* is. However, we know this not to be true since *timid* and *pushy* are antonyms and *timid* and *shy* are synonyms.

The reason this is a somewhat common occurrence is because GloVe uses words that are close in distance (within the document) to compute the similarity of meaning. It's the case that for some words, the antonym often appear very close which has the interpretation of having a close meaning despite us knowing this isn't the case. Maybe a better interpretation would be that words that are closer together are highly correlated, not necessarily similar in meaning. Although this generalization may be too broad to be useful and sort of defeats the purpose of us creating word embeddings in the first place.

## 2.4 Analogies with Word Vectors

Just to restate the variables, let  $m$  be a vector representing the word *man*,  $k$  be for *king*,  $w$  be for *woman*, and  $x$  be for the answer. All we're doing to

find the answer is subtracting  $m$  from  $k$  (trying to take the “male” out of king) and adding  $w$  (trying to add in female). We then find the word with the greatest cosine similarity to this vector.

## 2.5 Finding Analogies

This one is fun. I found that GloVe embeddings can recognize that *Lebron* is to *basketball* what *Brady* (Tom Brady) is to *football* which is a pretty accurate representation. They’re both star athletes in their respective sports and the fact we can identify what sport Tom Brady plays by comparing him to the sport that Lebron plays is pretty impressive.

## 2.6 Incorrect Analogy

The analogy I was looking for was: *rock* is to *hard* what *bed* is to *soft*. However, the analogy I actually get is: *rock* is to *hard* what *bed* is to *get*. What does that mean? Shit who knows.

## 2.7 Guided Analysis of Bias in Word Vectors

There’s a pretty clear gender bias between man in woman, particularly in that women are associated with being a nurse, teacher, or “mother” as a profession. Men on the other hand are associated with laborer (manual labor), mechanic, and factory. It’s interesting to note that men are associated with *working*, *job*, *unemployed*, etc. while women are associated much more with the home: *homemaker*, *child*, *pregnant*, etc.

## 2.8 Independent Analysis of Bias in Word Vectors

This one’s pretty interesting I think. You can see the bias between men and women in mathematics and sciences. When the analogy is of the form man:math :: woman:x, the answer is very much teaching related: graders, literacy, teacher, curriculum, kindergarten, etc. When the analogy is of the form woman:math :: man:x, the answer is quite different. We see words such as: whiz, genius, physics, chemistry, skills, etc. So this shows some clear bias between men and women in STEM.

## 2.9 Thinking About Bias

When we think about how these embeddings are developed, we're using the words often used with that word to indicate similarity. Thus, the embeddings reflect our own societal bias to talk about people a certain way. When we talk about men and mathematics we speak of geniuses, physics, and skill. When we talk about women and mathematics we speak of women teaching, grading, etc.

Obviously, the model itself isn't creating these biases. It's simply spitting back out the biases that currently exist in the data that it was trained on. This actually makes it a very interesting debugging tool for understanding the general trends and biases that exist within our language from a more computational and evidence based perspective. We're using math here, so the bias is in the data, more specifically our language and manner of speaking. This makes the biases we see in the model compelling evidence for how people generally view different people.

## 3 Resources

1. SVD Mining Massive Datasets Lecture 47- Stanford University (13 minutes)
2. CS 224N Lectures 1-2