TTIC 31190 HW1

Lynn Zheng

October 14, 2020

Note: `hw1.ipynb` contains all the code for this assignment. `distribution_counts.py` contains a modularized implementation of the function described in section 1.1. Design choices are briefly described in the notebook code and also discussed in the last section of this writeup.

1 Distributional Counting

1.1

For runtime performance (iterating through the corpus only once and computing counts for all three window sizes necessary for this assignment), the code in my notebook hard-codes the three windows and isn't very modularized. When initially developing the code, I wrote a more modularized function called `get_distribution_counts` in `distribution_counts.py`.

1.2

Word pair	$w = 3$	$w = 6$
#(chicken, the)	52	103
#(chicken, wings)	6	7
#(chicago, chicago)	38	122
#(coffee, the)	95	201
#(coffee, cup)	10	14
#(coffee, coffee)	4	36

Table 1: Counts for word pairs using $w = 3, 6$

1.3

(See section 4. Quantitative Analysis for the full table)

MEN: 0.225, SimLex-999: 0.0588

2 IDF

2.1

(See section 4. Quantitative Analysis for the full table)

Using IDF, we observe some improvement in the correlations.

MEN: 0.473, SimLex-999: 0.164

3 Pointwise Mutual Information

3.1

The PMI values are printed out in the notebook code file.

It's quite interesting to me how proper nouns like "costa" (Costa Coffee) and "Seattle" are found to be highly related to the word "coffee." We might also find "Starbucks" if we go down the list of words with large PMIs.

largest PMIs (largest to smallest)	smallest PMIs (smallest to largest)
tea	he
drinking	be
shop	had
costa	this
shops	not
sugar	its
coffee	after
mix	more
seattle	when
houses	page

Table 2: 10 context words with the largest/smallest PMIs for the center word "coffee"

3.2

(See section 4. Quantitative Analysis for the full table)

Compared to IDF, we observe some improvement in the correlation when evaluated on SimLex-999 but a slight decrease for MEN.

MEN: 0.466, SimLex-999: 0.186

4 Quantitative Comparisons

4.1

From the table below, the highest correlation (in boldface) is achieved on MEN using a context vocabulary of 5k, IDF with a window size of 6; the highest correlation achieved on SimLex uses a context vocabulary of 15k, PMI with a window size of 1.

The trends are pretty different for MEN and SimLex.

MEN: For each of the three methods, the correlation increases as the window size increases. For the **Counts** method, as the context vocabulary increases from 5k to 15k, the correlation decreases. For the **IDF** and the **PMI** method, as the context vocabulary increases, the correlation increases.

SimLex: For each of the three methods, the correlation decreases as the window size increases. For both the **Counts** and the **IDF** methods, the correlation decreases slightly as the context vocabulary increases from 5k to 15k. For the **PMI** method, as the context vocabulary increases, the correlation increases.

See below for the table, the plot, and my explanation/hypothesis for the trends.

vocab	method	window	MEN	SimLex
5k	counts	1	0.209	0.0678
		3	0.225	0.0588
		6	0.241	0.0447
	idf	1	0.348	0.189
		3	0.473	0.164
		6	**0.532**	0.111
	pmi	1	0.434	0.227
		3	0.466	0.186
		6	0.472	0.150
15k	counts	1	0.206	0.070
		3	0.221	0.0571
		6	0.237	0.0407
	idf	1	0.366	0.187
		3	0.481	0.148
		6	0.525	0.109
	pmi	1	0.470	**0.268**
		3	0.519	0.212
		6	0.527	0.161

Table 3: 36 correlations

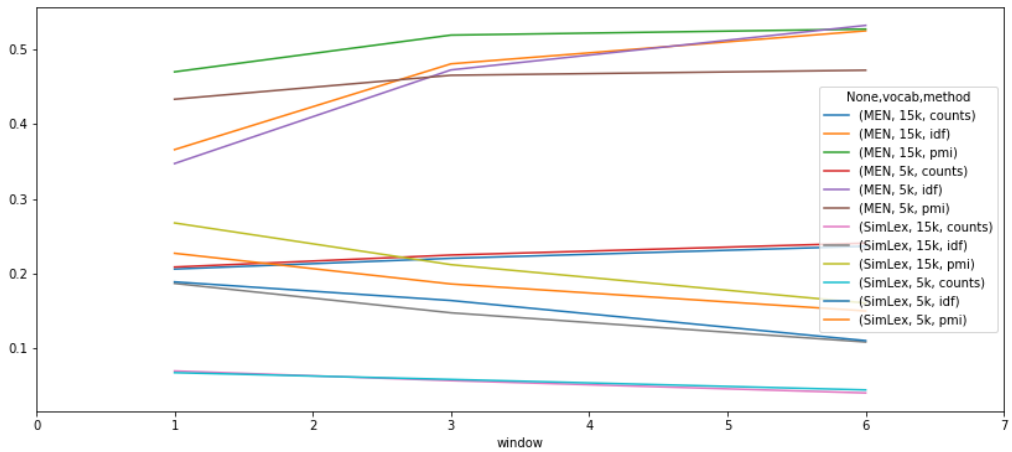


Figure 1: A plot of the correlations for MEN and SimLex, using $w = 1, 3, 6$

As window size increases for different method of creating word vectors, I'd expect the correlations to increase. This is because a large enough (but not too large) window size like 6 should give us just enough context. (A window as narrow as 1 and 3 may limit the amount of available context information, while a window too wide may introduce noisy, irrelevant context.) The trend we observe on MEN aligns with my hypothesis.

As context vocabulary increases, I'd expect correlation to also increase. This is because more words inside the window surrounding the keyword are now considered context word, effectively increasing the amount of context information available to us for determining word similarity. My observation confirms my hypothesis: for both MEN and SimLex, using IDF and PMI, for the most part, correlation increases as context vocabulary increases. This trend is not apparent for raw distributional counts because with a larger context vocabulary, we may be counting many more common words like "the" which confounds our computation of meaningful keyword-context counts.

4.2

The two datasets are encoding different types of similarity.

MEN encodes contextual similarity. Word pairs that people would associate together tend to get high scores even if they have different part of speech, like "music" and "sing", "beach" and "swimming". Objects frequently occurring together in daily life also have high scores, like "car" and "garage".

SimLex encodes synonymity. Synonymous word pairs tend to score high, like "hard" and "difficult" with a high score of 8.77, whereas antonymous word pairs tend to score low, like "easy" and "difficult" with a low score of 0.58. The word pairs in this dataset also usually have the same part of speech, so it's difficult to tell how SimLex will rank pairs with the same stem but of different part of speech like "speak" and "speech".

A good example contrasting the notion of similarity between the two datasets: The pair "mother" and "son" has a rather high similarity of 41 in MEN, but the pair "father" and "son" only has a medium similarity of 3.82 in SimLex. As another example, "dinner" and "breakfast" only scores a medium 3.33 in SimLex but could have scored much higher in MEN as both words are related to meals.

5 Qualitative Analysis

To visualize my word embeddings, I selected some words and uploaded their vectors to the TensorFlow TensorBoard project. An initial look at the visualization shows that words and their nearest neighbors don't necessarily share the same part of speech or word stem. Please see section 6. Visualization for a detailed explanation about my TensorBoard configuration.

5.1

$w = 1$	$w = 6$
judge	judge
justices	appeals
arbitrators	supreme
players	court
trustees	panel
contestants	courts
officials	jury
admins	contestants
appeals	justice
officers	officials

Table 4: 10 nearest neighbors for "judges" using $w = 1, 6$

5.2

For some selected nouns, verbs, adjectives, and prepositions, highlighted especially interesting ones in boldface and took notes in the **Note** column of each table.

(Query words that have almost exactly the same nearest neighbors with the two window sizes are shown in boldface below) Nouns: speech, **window**, neighbor, success
Verbs: climbed, **speaks**, **spoke**
Adjectives: sunny, happy, unfortunate
Prepositions: about, between, within

It appears that unambiguous words, concrete nouns (as opposed to abstract nouns) might have similar nearest neighbors for the two different window sizes.

A systematic pattern between the two different window sizes, especially for some verbs and adjectives, is that $w = 1$ seems to find neighbors with similar part of speech and contextual meaning; In comparison, $w = 6$ seems to find neighbors that share mutual context information (as expected from PMI) and are the subjects or objects the keyword interact with (Please suggest some more rigid way to characterize this relationship), but not necessarily of the same part of speech. This difference is most evident for verbs like **climbed**, adjectives like **happy, unfortunate** and the preposition **within**.

Noun	$w = 1$	$w = 6$	Note
speech	voice rf statement action **address**	freedom voice communication ideas expression	**address** here is multi-sense, as in giving a **speech**
window	windows doors door tower panel	windows door roof floor glass	
neighbor	classmates grandparents willingness brother-in-law friend	mrs partner **lucy** girlfriend arrives	**Lucy** might just be a very common name
success	popularity successes interest impact acclaim	successful popularity **hit** despite winning	**hit** here is multi-sense and synonymous to *success*

Verb	$w = 1$	$w = 6$	Note
climbed	travelled sailed flown pushed shipped	reaching billboard climb charts climbing	For $w = 1$, the nearest neighbors are all inflected verbs
speaks	spoke speak preached cared fluent	speak spoke knows sees speaking	
spoke	speak speaks spoken knew disagreed	speak speaking spoken speaks speech	For $w = 6$, the 5 nearest neighbors all stem from **speak**

Adjective	$w = 1$	$w = 6$	Note
sunny	elevator rainy **köppen** **humboldt** rocky	dry moist rainy humid wet	Köppen is a climate classification system Humboldt is a county in California
happy	pleased surprised worried glad sorry	anyone 'll everyone 'd let	For $w = 1$, the neighbors are adjectives describing feelings For $w = 6$, the neighbors look like they are from sentences like "They'll let everyone be happy"
unfortunate	tragic annoying sad touching painful	obvious **rfa** **admins** terrible **admin**	For $w = 1$, we have adjectives with negative sentiments RFA could possibly mean Wikipedia "Requests for adminship" as **admin** is also in the list. No ideas about why **admins** are so **unfortunate**, though.

Preposition	$w = 1$	$w = 6$	**Note**
about	over than like years if	i not this there that	
between	until around since through october	south north in from east	
within	across around throughout among along	area areas small large region	For $w = 1$, all neighbors are prepositions. For $w = 6$, three are nouns and two are adjectives.

5.3

For words with multiple senses, it appears that usually one sense dominates the other, resulting in neighbors similar to that particular sense: Examples include **cell** and **well**. Interestingly, with different window sizes, different senses may dominate, as in the case of **apple**. When this happens, the smaller window size $w = 1$ usually results in a dominant sense with more concrete meaning, whereas the larger window size $w = 6$ results in one with more abstract meaning. This may be that with more context, it is easier to find/disambiguate the abstract meanings of certain nouns.

Multisense	$w = 1$	$w = 6$	Note
bank	side coast railway park africa	capital corporation railway northern branch	For $w = 1$, the dominant sense seems to be “river side” For $w = 6$, the dominant sense is the “financial institution”
cell	cells tissue tissues human brain	cells protein function dna surface	The biological sense dominates the “prisoner room” sense
apple	pine atari cherry christmas olive	microsoft computers os desktop mac	Atari is a game and computer company For $w = 1$, the dominant sense is the fruit For $w = 6$, the dominant sense is “Apple, Inc.”
apples	tomatoes flowers impatient grapes guys	fruits grapes fruit vegetables wheat	Unambiguously, the plural form refers to the fruit
axes	tributaries branches phases viewpoints dimensions	parallel horizontal angles strings axis	$w = 6$ focuses more on its mathematical sense
frame	wooden brick two-story framed frames	roof rear wooden structure brick	The architectural sense dominates
light	**heavy** water **dark** line large	**heavy** surface **dark** color body	For both window sizes, two dominant senses “weighing little” and “bright” and their corresponding antonyms emerge
well	however united preserved list discussion	such other many most are	The adverb of “good” dominates the water excavation site

6 Visualization with TensorBoard

I created word vectors with IDF and PMI, $w = 6$, and 5k context vocab (5k instead of 15k for a smaller file size). Link to the visualization. I only created 40 - 50 word vectors for each experimentation due to the limitation of file sizes that I can host.

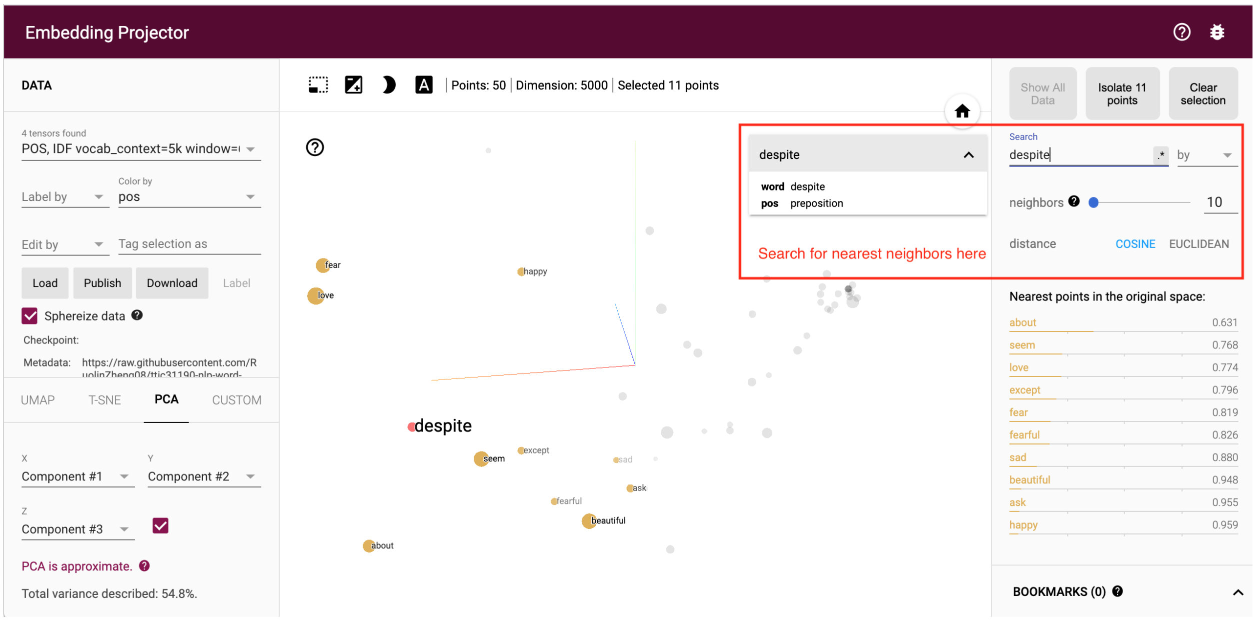
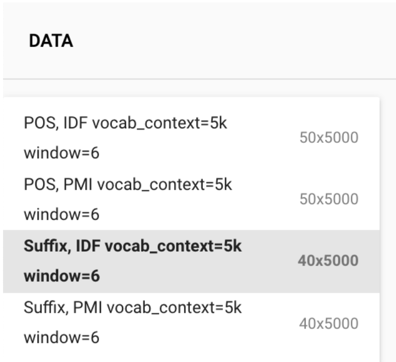


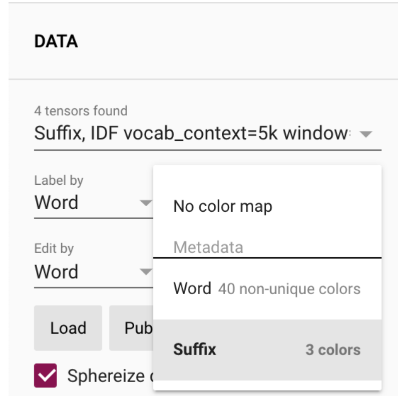
Figure 2: TensorBoard UI to compute nearest neighbors

I labeled one set of words by their part of speech (POS) and another set by their suffix (one of "ly", "tion", "ing", despite some noise like "family", this almost corresponds to adverbs, nouns, and verbs). From the visualization, words with the same labels don't necessarily cluster together in space (using PCA in three dimensions by default, but I tried t-SNE and it didn't do much better). I got the inspiration to use suffixes from my TTIC Speech Technologies class project where we trained acoustic word embeddings using neural methods and obtained good clustering by suffixes.

To toggle between the four experimentation sets and color the points by POS or suffixes:



(a) Toggle between embedding sets



(b) Toggle colormap

7 Design Remarks

My code takes 10 minutes to construct all 18 maps (3 methods, 3 window sizes, 2 vocabs) used to create word embeddings.

The extensive use of the `set` data structure makes the code incompatible with Python 2. When constructing the word vectors given a context vocabulary, we iterate over a set, and, unlike in Python 3, in Python 2, iterating over a set is non-deterministic. Therefore, the ordering of the context vocabulary might differ for multiple calls to the function to construct a word vector. To make the code compatible for Python 2, the function to compute word vectors needs both a vocab set (for constant-time lookup) and a vocab list (for deterministic ordering).