Distributional Semantics Takes the SAT

CS114 (Spring 2020) Programming Assignment 5

Crystal Lee

04/29/2020

1. Create distributional semantic word vectors

Q1: Compare the word vector for "dogs" before and after PPMI reweighting. Does PPMI do the right thing to the count matrix? Why? Explain in a few sentences how PPMI helps.

	feed	women	dogs	like	bite	the	men
Raw Counts	1	1	1	1	1	91	1
PPMI	0	0	0	0	0	2.09	0

Table 1.

Yes, PPMI indeed improves the measurement of the association between words. Instead of directly using raw counts, it captures how much more the two words co-occur than two words appear by chance. Therefore, it can discriminate against essential words.

As we can see in **Table 1**, in the raw count vector, the context word, "the", has the highest frequency, and its value is also other than 1 compared to other context words. The PPMI vector has similar results, and only "the" has a non-zero value. However, the range between "the" and other context words become smaller in the PPMI vector. Therefore, PPMI can not only correctly discriminate important words but also mitigate a skewed problem of raw counts. Because "the" are ubiquitous in all sentences, the prior probability of "the" will be high, and the independent probability of "dogs" and "the" will also be high. Thus, the importance of "the" will significantly decrease. Although "the" is not the right word to differentiate "dogs" with other nouns, it can help us to discriminate it with other parts-of-speech, such as verbs. That's the reason why "the" still has a certain amount of PPMI.

Q2: Do the distances you compute above confirm our intuition from distributional semantics (i.e. similar words appear in similar contexts)?

Yes, the distances reflect the fact that similar words appear in similar contexts(**Table 2**). The first three pairs are all nouns and have similar distances because their contexts are the same word, "the". Although "men" seems more closer to "dogs" than to "women", it might result from the difference of the total frequency in our dataset. If we compute how many times those nouns are shown in our dataset, we can discover that "dogs" appears 91 times, "men" occurs 81 times and "women" only shows 51 times. Thus, we can conclude that the differences of distance here between nouns are determined by how often a word appears in all sentences.

Although human-related and animal-related nouns don't have a significant difference, we can see the apparent disparity between human-related and animal-related verbs. Since both "feed" and "like" belong to human-related verbs, they should be close to each other but be far from the animal-related verb. This fact is revealed by the above measurement of distance.

	pairs	compact PPMI
0	women_men	0.2234
1	women_dogs	0.3398
2	men_dogs	0.1164
3	feed_like	0.6674
4	feed_bite	2.1746
5	like_bite	1.7205

Table 2.

Q3: Does the compact/reduced matrix still keep the information we need for each word vector?

Overall, the reduced PPMI remains most information for each word vector(**Table 3**). Although only the distance between "feed" and "like" decreases slightly, it still reflects that "feed" and "like" are closer to each other and don't influence the results.

pairs	compact PPMI	reduced PPMI
women_men	0.2234	0.2234
women_dogs	0.3398	0.3398
men_dogs	0.1164	0.1164
feed_like	0.6674	0.522
feed_bite	2.1746	2.1701
like_bite	1.7205	1.6985

Table 3.

2. Synonym Detection

Obviously, Euclidean distance performs better in synonyms detections than cosine similarity(**Table 4**). The former can correctly answer the question and gain 100% accuracy. As for COMPOSES and word2vec, if we look at the performance of cosine similarity, the Google matrix(word2vec) seems to have relatively better performance than the classic matrix(COMPOSES). (Accuracy: 62.7% > 53.7%)

	Accuracy
cos_google	0.627
cos_classic	0.537
dist_google	1
dist_classic	1

Table 4.

3. SAT Analogy questions

To obtain a relation between two words within a question and its choices, I tried several methods, including division, multiplication, addition, concatenation, and subtraction. Subsequently, I compared those relations through computing their cosine similarity and Euclidean distance and chose the maximum value for cosine similarity and the minimum values for Euclidean distance. Besides, each method adopted two kinds of word matrices, the Classic matrix (COMPOSES) and the Google matrix (word2vect), respectively, to conduct computation. Finally, by calculating the accuracy for each method, the results are shown in **Table 5**.

Similarity	Aggregation Method / Word Matrix	Division	Multiplication	Addition	Concatenation	Subtraction
Cosine Similarity	Classic	0.1818	0.2353	0.3235	0.3904	0.4225
	Google	0.1813	0.2373	0.3253	0.3920	0.4213
Euclidean Distance	Classic	0.2032	0.2594	0.3102	0.3904	0.3824
	Google	0.2053	0.2587	0.3120	0.3920	0.3840

Table 5.

In order to clearly identify the best method, I visualized the results through the parallel coordinate plot and presented it in **Figure 1**. Each vertical line represents the method for computing a relation between two words. Color lines exhibit the accuracy of different similarity methods. Red lines are cosine similarity whereas blue lines are Euclidean distance.

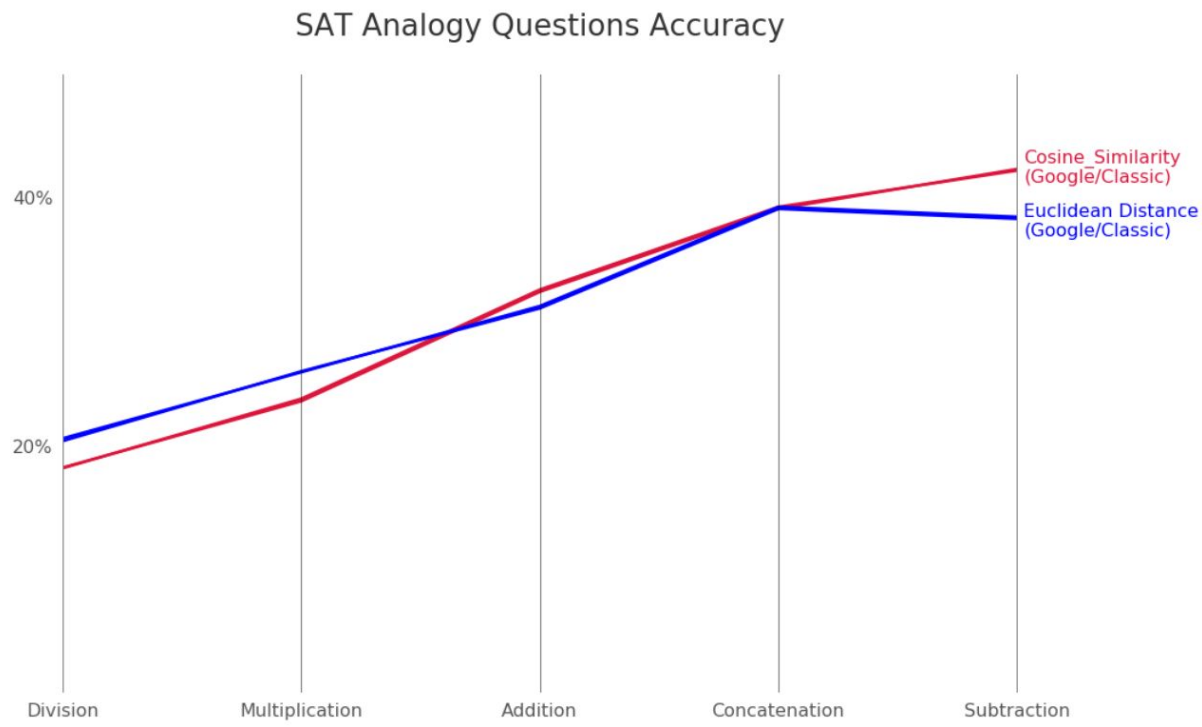

Figure 1.

We can notice that the best combination is to use the subtraction method and cosine similarity. For analogy questions, there is somehow a correlation between two words, but we might not be able to describe it as a positive or negative relationship. Take a question, "ostrich" and " bird" as an example. "Ostrich" is one kind of "birds''. That is, "ostrich" belongs to the "bird" category. Therefore, the best answer is "lion and cat" because "lion" can also be classified into the "cat" category. The relationship between two words is that one word is a general concept, and the other word is a specific example of the concept. To capture this kind of relation, we can compute how much difference two word vectors have through the subtraction method. Besides, when using the subtraction method, the cosine similarity method has a better effect than Euclidean distance. Because subtraction summarizes a relation of two words into a vector and keeps a property of direction. The cosine similarity can recognize whether two vectors have a similar direction, but Euclidean distance only identifies the distance between two vectors. However, when using the concatenating method, because we kept all information from two words, Euclidean distance and cosine similarity had equivalent performances. Therefore, different aggregating methods should combine different similarity methods to optimize the performance

As for the word matrices, the Classic matrix has similar performance with the Google matrix in no matter what method we use. For the best method (subtraction and cosine similarity), the Classic matrix slightly better than the Google matrix (0.4225 > 0.4213). Overall, the numbers of entries in a vector will not cause a significant influence on accuracy.