Carmi Rothberg

May 7, 2017

CS114 (Spring 2017) Homework 5

Distributional Semantics Takes the SAT

PART I: Building Vectors

After we build our initial co-occurrence matrix *C*, we reweight it with the PPMI. This has the following effect:

Before PPMI reweighting, the word vector for 'dogs' is:

women	bite	the	men	feed	dogs	like
0.0	37.88079	0.63130	0.0	0.0	0.0	0.00688

After PPMI reweighting, the word vector for 'dogs' is:

women	bite	the	men	feed	dogs	like
0.0	37.28603	0.20358	0.0	0.0	0.0	0.00128

The difference after PPMI reweighting is relatively small, but it seems to be doing about the right thing. The co-occurrence of ‘dogs’ with ‘bite’ is emphasized, and words that occur less frequently with ‘dogs’ — such as ‘the’ and ‘like’ — receive lower values. This works with our intuitive semantic understanding of ‘dog’—that dogs are associated specifically with biting — as is reflected in the data — where biting is always and only occurs in sentences with dogs.

Using the semantic model we've just created, we may now calculate the Euclidean distances for the following word pairs:

```
The Euclidean distance between "women" and "men" is 34.99621.
The Euclidean distance between "women" and "dogs" is 73.52287.
The Euclidean distance between "men" and "dogs" is 58.22367.
The Euclidean distance between "feed" and "like" is 23.92139.
The Euclidean distance between "feed" and "bite" is 61.28419.
The Euclidean distance between "like" and "bite" is 54.36544.
```

These distances confirm our intuition from distributional semantics — words like ‘men’ and ‘women’, which are similar in meaning, appear in similar contexts — far more similar contexts than ‘men’ and ‘dogs’, for example, or ‘women’ and ‘dogs’.

Using SVD, we may reduce the matrix to take up considerably less space — a reduction from 300,000 into 500. Importantly, this reduction does not lose us much information. Even in the compact matrix, though the values themselves shift around a bit, the general trends in values remain the same, and we preserve the essential information for each vector. Note the new Euclidean distances between the following word pairs:

```
The Euclidean distance between "women" and "men" is 28.97929.
The Euclidean distance between "women" and "dogs" is 25.14256.
The Euclidean distance between "men" and "dogs" is 5.23755.
The Euclidean distance between "feed" and "like" is 11.48276.
The Euclidean distance between "feed" and "bite" is 40.99892.
The Euclidean distance between "like" and "bite" is 36.29372.
```

PART II: Analyzing and Applying Vectors

Let's now take a look at the variety of methods to solve the synonym problems.

First, we may try each vector data set — the news data from the COMPOSES and word2vec toolkits — with a Euclidean distance measure. This nets us 658 correct out of 1000 — 66% — with both sets of vector data.

With cosine similarity, however, the Google data is slightly more effective than the COMPOSES data — in fact, the COMPOSES data seems to be less reliable in the cosine similarity test than with Euclidean distance. The COMPOSES data results in 619 correct (62%), while word2vec answers 734 correct (73%). (Note in the code that cosine distance was used instead of cosine similarity in order to allow for the reuse of code from the Euclidean measure. However, the process of calculating the distance is nearly identical to the process of calculating the similarity — cosine distance is merely 1-cosine similarity.)

RESULTS FOR SYNONYM TESTS	
Euclidean Distance	Cosine Similarity
66%	62%, 73%

In the SAT problems, the best results I achieved hovered around 29-30%, using cosine distance and the COMPOSES data. The results are presented in the following tables:

My first try for the analogies was to use multiplication and division — for example, in an analogy `A:B :: 1:2`, I would calculate `(cosine(v2, v1) * cosine(v2, vB)) / cosine(v2, vA)` for each pair of 1 and 2 in the five choices and pick the pair that resulted in the maximum cosine similarity / minimum cosine distance. However, this proved to have comparatively poor results (see below), so I decided instead to use addition and subtraction, as in the class demonstration with the man-woman|king-queen analogy. Specifically, I calculated `cosine(v2, v1) - cosine(v2, vA) + cosine(v2, vB)` for each word pair.

SAT RESULTS FOR VECTOR MULTIPLICATION		
	Euclidean Distance	Cosine Distance
Google Data	89 correct out of 374 (23.8%)	96 correct out of 374 (25.6%)
COMPOSES Data	89 correct out of 374 (23.8%)	109 correct out of 374 (29.1%)

SAT RESULTS FOR VECTOR ADDITION

	Euclidean Distance	Cosine Distance
Google Data	89 correct out of 374 (23.8%)	101 correct out of 374 (27.0%)
COMPOSES Data	89 correct out of 374 (23.8%)	111 correct out of 374 (29.7%)

NOTES:

The synonym test set can be seen in full by typing `print(problems)` after running the Assignment 5 code. Here is a sample:

```
[['to_swat', 'to_paste', 'to_cut', 'to_misunderstand', 'to_pollute', 'to_pare'],
 ['to_bulge', 'to_expand', 'to_open', 'to_hit', 'to_classify', 'to_centralize'],
 ['to_confine', 'to_bind', 'to_spin', 'to_like', 'to_be_formed', 'to_close'],
 ['to_whisk', 'to_stir', 'to_add', 'to_lecture', 'to_sparkle', 'to_inspect'],
 ['to_boom', 'to_bang', 'to_hurt', 'to_offend', 'to_file', 'to_weaken'],
 ['to_arrive', 'to_reach', 'to_stop', 'to_consolidate', 'to_reproduce', 'to_imagine'],
 ['to_organise', 'to_arrange', 'to_massage', 'to_weave', 'to_labor', 'to_let_escape'],
 ['to_redirect', 'to_change_course', 'to_centralize', 'to_copy', 'to_weaken', 'to_group'],
 ['to_edit', 'to_change', 'to_whine', 'to_dislocate', 'to_duplicate', 'to_hop'],
 ['to_arrive', 'to_land', 'to_void', 'to_bound', 'to_divide', 'to_throw']]
```

Test sets and generated solutions for other problems are also stored by the submitted code and should be viewable if necessary.