CSC2611 Lab (Bai Li)

The Github repo is available at: https://github.com/luckytoilet/csc2611-semantic-change. However, due to academic integrity concerns, I have set the visibility to private. If the instructor wants to view the repo, I can give you access by adding you as a collaborator.

Exercise

I got the following results for Pearson correlation with human similarity judgements:

- M1: 0.216
- M1P: 0.415
- M2_10: 0.133
- M2_100: 0.329
- M2_300: 0.411

The Brown corpus suffers from data sparsity issues: some words in the RG65 table appear very few times, like "madhouse" and "woodland". Thus, for similarity calculations, I discarded words that appear in fewer than 3 bigrams. Out of the 65 pairs in the table, we are left with 40 remaining to calculate Pearson correlation.

The PPMI embeddings (M1P) perform the best, followed closely by the 300 dimensional SVD embeddings (M2_300).

Part 1: Synchronic word embeddings

Step 3. Using word2vec embeddings, I get Pearson correlation of 0.772 with human similarity judgements. This is higher than 0.411 from the exercise using LSA vectors.

Step 4. We only consider analogy cases where all 4 words are in the LSA vocab; this leaves us with 2207 tests. Out of these, 162 are semantic and 2045 are syntactic.

For each tuple w_1, w_2, w_3, w_4, we evaluate the models by taking computing the vector for $w_4' = w_3 + w_2 - w_1$ and retrieving word with vector closest to w_4' by cosine distance. We exclude $\{w_1, w_2, w_3\}$ from consideration. The test case is counted as correct if $w_4' = w_4$.

Table of results:

	W2V	LSA
All	1599/2207 (72%)	225/2207 (10%)
Semantic	112/162 (69%)	19/162 (12%)
Syntactic	1487/2045 (73%)	206/2045 (10%)

Word2vec produces much better word analogy performance than LSA. This is not too surprising because the LSA model was trained on only about 1M words in the Brown corpus, compared to the GoogleNews vectors, which is trained on 3B words. A fairer evaluation should use the same training data for both models.

Step 5. There are numerous ways to improve on word2vec and LSA.

First, both models treat each word as a unique identifier, which doesn't capture the morphology within a word. For example, "*walk*", "*walks*", "*walked*" are treated as three separate words. A better method should capture the morphological relationships between this words, for example, *"walks"* = *"walk"* + *"s"*. Subsequent models like FastText used subword embeddings to better handle morphology, especially in uncommon words.

Second, word2vec and LSA are poor at handling homonymy and polysemy, where the same word has different meanings depending on context. For example, the noun meaning and verb meaning of "*bear*" are completely unrelated. One way this may be improved is by running a part-of-speech tagger as preprocessing, and treat each instances of a word with different part-of-speech tags as different items. Thus, "*bear.N*" and "*bear.V*" would be different words. This would not eliminate homonymy problems completely, as some words like "*bank*" may have multiple senses that are all nouns.

Part 2: Diachronic word embeddings

Step 2. We try the following methods to measure semantic change for a word:

- **FIRST**: take the cosine distance between the first embedding of a word (1900) and the last embedding of a word (1990).
- **MAX**: take the maximum of the pairwise cosine distances for all the embeddings of a word.
- **SUM**: take the sum of cosine distances of consecutive word embeddings for a word.

Most changing for **FIRST**: programs, computer, radio, approach, patterns, signal, levels, project, league, pattern, technology, content, post, economy, program, t, evaluation, jobs, bit, model

Least changing for **FIRST**: autumn, clergy, villages, commodities, newspapers, vicinity, remark, votes, fleet, priest, poets, drama, symbol, prosperity, temperatures, defeat, seas, colonel, phrase, allies

Most changing for **MAX**: objectives, computer, programs, sector, radio, goals, perspective, shri, impact, approach, van, media, patterns, assessment, berkeley, princeton, shift, therapy, film, j

Least changing for **MAX**: april, november, february, october, january, june, december, september, century, daughter, evening, july, husband, coast, trees, river, church, increase, god, miles

Most changing for **SUM**: programs, technology, time, example, smith, peter, way, course, instance, case, texas, league, robert, columbia, box, fact, thomas, bay, others, grand

Least changing for **SUM**: dress, acres, diagnosis, crops, imprisonment, quantities, exports, affection, dioxide, ranks, autumn, vicinity, nerves, rooms, completion, colour, customs, hearts, symbol, confusion

Pearson correlations:

	FIRST	MAX	SUM
FIRST	1	0.98	0.68
MAX	0.98	1	0.72
SUM	0.68	0.72	1

The results look reasonable: all methods identified technology-related words like "*computer*" and "*program*" as semantically changing. There are a few nonsensical words like "*shri*" and "*j*" and I'm not sure why they're contained in the list of 2000 words at all.

Step 3. Following Xu and Kemp (2015), we compute the ground truth for a word w by taking the 100 words closest to w in 1900 and the 100 closest words in 1990, and define the semantic change to be $1 - \frac{overlap}{100}$.

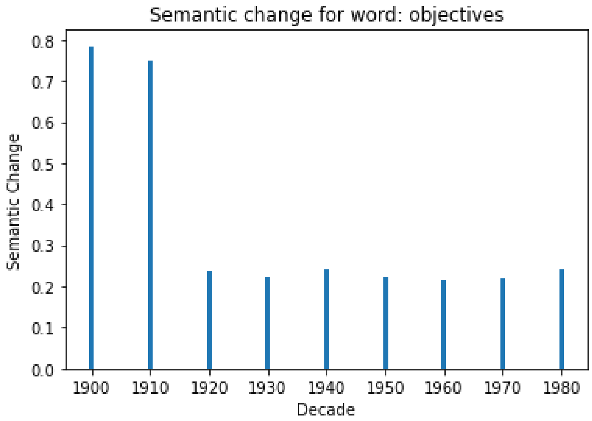
Pearson correlations for each method:

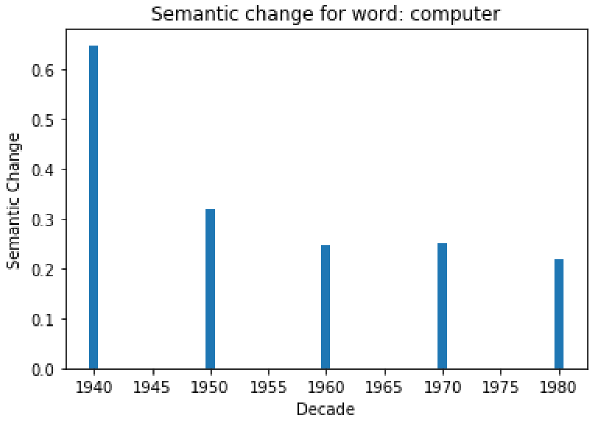
- FIRST: 0.410
- MAX: 0.414
- SUM: 0.236

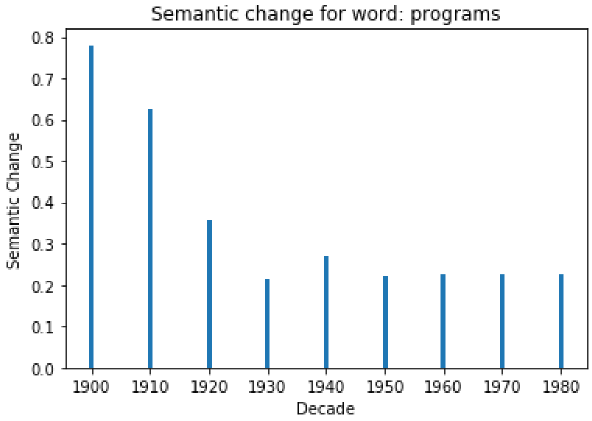
The MAX method has highest evaluation accuracy.

Step 4. Recall from step 2 that the top 3 changing words according to the MAX method are: *objectives, computer, programs*.

We identify the change point by taking the maximum cosine distance between two consecutive timesteps.







According to these graphs, we can see that *objectives* changed the most in 1900-1910, *computer* changed most in 1940-1950, and *programs* changed most in 1900-1910.

The results should be taken with a grain of salt: according to Google ngrams, *programs* is virtually unattested in the year 1900, and *computer* is first attested during the 1940s. Thus the word embeddings likely have high variance for the periods during which the word is rare. To fix this problem, it would be worth investigating methods that are weighted by frequency of usage; however, frequency information is not provided in the dataset.