CSC2611 Lab Assignment

Tony Nguyen

February 4th, 2020

1 Synchronic word embedding

3. Cosine similarities:

```
cord; smile: 0.018116442126818393
rooster; voyage: 0.06275809503788346
noon; string: 0.02165451453599645
fruit; furnace: 0.07321497303845254
autograph; shore: 0.034655916200793474
automobile; wizard: -0.02808742179782148
mount; stove: 0.049256221719463845
grin; implement: -0.00023054715138644667
asylum; fruit: 0.05780962502651894
asylum; monk: 0.13866824166204164
graveyard; madhouse: 0.29396817045820234
glass; magician: 0.037224910380892116
boy; rooster: 0.2848518997190719
cushion; jewel: 0.12478441292362313
monk; slave: 0.19146227120982814
asylum; cemetery: 0.09243521370466698
coast; forest: 0.23609790007924333
grin; lad: 0.2480188694136008
shore; woodland: 0.11690946166507768
monk; oracle: 0.30354036186223615
boy; sage: 0.16595692925794378
automobile; cushion: 0.13358584792553976
mound; shore: 0.13164130789280562
lad; wizard: 0.33023006071411043
forest; graveyard: 0.22901121328397625
food; rooster: 0.11830646680620911
cemetery; woodland: 0.38192504756514994
shore; voyage: 0.20434848891292628
bird; woodland: 0.3402425169680229
coast; hill: 0.16115775478766303
furnace; implement: 0.023429482361675774
crane; rooster: 0.2360721403228397
hill; woodland: 0.27359094869895173
car; journey: 0.09849625033300127
cemetery; mound: 0.20604093400134227
glass; jewel: 0.17449359032063172
magician; oracle: 0.25220774378294275
crane; implement: 0.023186153923084922
brother; lad: 0.35959248937893756
sage; wizard: 0.338115043980551
oracle; sage: 0.44257072376116036
bird; crane: 0.30286191057859796
bird; cock: 0.36290242005508533
food; fruit: 0.3740925987787312
brother; monk: 0.22320019970585045
asylum; madhouse: 0.25253930133545643
furnace; stove: 0.6083910645324063
magician; wizard: 0.4863496163040532
hill; mound: 0.4622032618800124
cord; string: 0.18951252054609116
glass; tumbler: 0.46751660119705624
grin; smile: 0.8604010001094325
serf; slave: 0.44984420342273834
journey; voyage: 0.6830852596999198
autograph; signature: 0.3132111796287731
coast; shore: 0.5083667401271162
forest; woodland: 0.6417990211906214
implement; tool: 0.21234227421379986
cock; rooster: 0.47867878710642686
boy; lad: 0.5886159347560659
cushion; pillow: 0.2516151650781008
cemetery; graveyard: 0.6424806430243765
automobile; car: 0.5838367575984194
midday; noon: 0.552740644148218
gem; jewel: 0.6210810621214565
```

Pearson correlation between word2vec-based similarities and human similarities: `0.7838683898261164`

From the exercise, the Pearson correlation coefficient between similarities rated by humans and those computed from LSA word embeddings with full size, 300, 100 and 10 dimensions are `-0.010896583768295408`, `-0.007959273781947773`, `0.016467882646819598`, `0.010170006153905543`, respectively.

Mathematically, the Pearson correlation coefficient between 2 samples are in the range $[-1, 1]$, where values closer to -1 indicates a negative correlation whereas values closer to 1 indicates a positive correlation. Values close to 0 indicates that there is little to no correlation.

We see above that the Pearson correlation coefficients between human similarity scores and LSA embeddings similarity scores are very close to 0, and thus indicate that the similarities computed by using cosine distance on the LSA embeddings do not well-represent the semantic distance of these words in accordance to human standards. On the other hand, the Pearson correlation coefficient between the similarity scores computed using word word2vec embeddings and those proposed by humans is closer to 1, hence, suggesting that this similarity computation performs well at estimating the semantic distance between words.

4. To account for the fact that the LSA embeddings can only work with a small subset of the testing data, I extracted only those testing examples where all 4 words are contained within the LSA and word2vec vocabulary. The accuracy of the analogy test using word2vec embeddings is `68.5%`, compared to the accuracy of `0%` using the LSA word embeddings.

This result is reasonable as the LSA embeddings simply capture the meaning of a word via its co-occurrence information with other words in the vocabulary, which does not necessarily establish intimate semantic connections. In contrast, the word2vec model is trained using the continuous skip-gram objective, and is a prediction-based model, it learns the embeddings by predicting a context window given a target word, thus, given a large amount of data, can learn the fine-grained relationships between words. The word2vec model used in this assignment is trained on 100 billion words, which is a very sizable dataset, and thus outperforms the LSA technique. Furthermore, we limit our LSA computations to only use the top 5000 most frequent words, and thus the word embeddings would not be as expressive.

5. A popular technique that has emerged in recent years in natural language processing is the use of attention. Attention allows a neural model to weigh the context words of a target word differently due to a compatibility measurement (e.g. dot-product). This technique has proven to be effective as it has beat out previous solutions without it, and produce state-of-the-art results in downstream NLP tasks. A way of increasing the quality of these vector-based models is to apply the attention mechanism when performing the objective, and thus allows the task to produce better results, thus the weight matrix that is used to create the word embeddings will also be of higher quality.

2 Diachronic word embedding

2. In this part, we will use the following 3 metrics of measuring degree of semantic change for a word between time steps t and t' using:

- cosine-distance: we compute the cosine-distance between the vector representation of this word in time t and time t'.
  $\rightarrow$ 20 most changed:

mcgraw	objectives	approach	perspective
skills	computer	van	patterns
ml	radio	shri	berkeley
techniques	sector	media	shift
programs	goals	impact	film

  $\rightarrow$ 20 least changed:

april	october	daughter	week
june	increase	december	evening
november	january	god	door
february	century	september	payment
years	months	feet	miles

- average between k nearest neighbors: here, we take the k (set to 100 in this analysis) nearest neighbors to the word in time t and we do the same for time t'. We take the average across each dimension between all neighbors in time t and time t' and compute the cosine-distance between these 2 vectors.
  $\rightarrow$ 20 most changed:

ml	sector	center	programs
approach	computer	release	techniques
radio	film	host	skills
signal	impact	media	focus
objectives	mcgraw	goals	assessment

→ 20 least changed:

profits	purchase	assets	fluid
thomas	coast	oxygen	acid
taxes	increase	shares	money
payment	acids	loan	desert
velocity	san	richard	increases

- Euclidean distance: we compute the Euclidean distance between the word's embedding in time t and time t'.
  → 20 most changed:

therapy	shift	media	sector
challenge	berkeley	shri	radio
stanford	patterns	van	computer
assessment	perspective	approach	objectives
film	impact	goals	programs

→ 20 least changed:

programs	goals	impact	film
objectives	approach	perspective	assessment
computer	van	patterns	stanford
radio	shri	berkeley	challenge
sector	media	shift	therapy

The intercorrelations:

	cos	knn	euc
cos	1	0.51203498	1
knn	0.51203498	1	0.49466547
euc	1	0.49466547	1

3. To quantify the accuracy of the measurements above, I found a small dataset found here. This is the dataset that was used in **Statistically Significant Detection of Linguistic Change** by Kulkarni et al. The authors extracted what they deemed to be the top 20 most semantically changed words and ask 3 people to evaluate whether or not the word has changed in meaning over time significantly (answer 0 or 1). I have compiled all 3 answers for each word and taken the average of the 3 answers to get the human perceived semantic change. I then computed the Pearson correlation coefficient between the semantic change computed by our methods for words that are both in our vocabulary and the top 20 extracted words by the authors. The correlation score is the accuracy as it measures how much it matches the human evaluation, if the Pearson correlation coefficient of any method and the human scores is negative, it is taken to be 0 accuracy, as the method's results negatively correlates with the "expected" results. The accuracies are as follows:

```
Cosine-distance method: 0.43441324175136753
Average kNN cosine-distance method: 0.20449106250538848
Euclidean distance method: 0.43441324175136803
```

Ideally, there would be a bigger dataset for the evaluation of the metrics aforementioned to be more reliable.

4. From the previous part, we see that the best performing metric is the Euclidean distance measurement. Using this, the top 3 most changed words throughout the 10 decades are: `programs`, `objectives` and `computer`.

A simple way of detecting change point(s) is to iterate through each decade, with the initial point of reference being the first decade. At each decade, we retrieve the computed semantic change degree between the decade in question and the reference point and compare this value against a threshold that is chosen. If the change value is at least the threshold, we denote the decade to be a change point, and update the reference point to be the current decade.

Change over time plots and change points (next page):

In the plots below, the list at the bottom of the graph shows the detected change points. The graph has on the x-axis the decades and the y-axis the Euclidean distance between the vector of the word at the corresponding decade and the latest reference point. For decade d, the latest reference point is the maximum of all reference points that is less than d. The threshold was set to **0.4**.

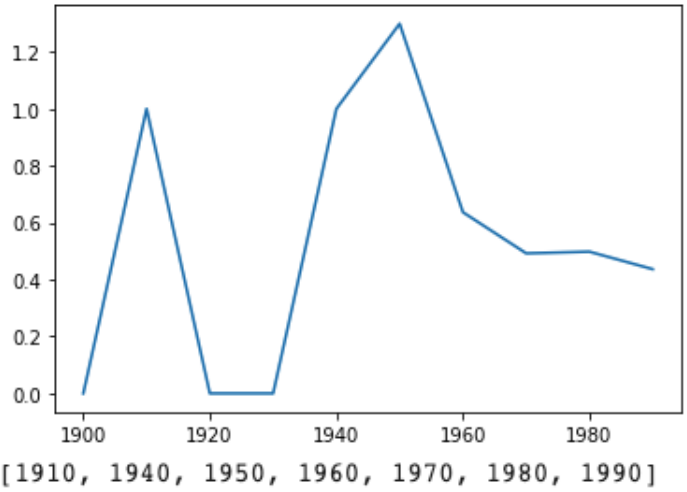

Figure 1: Distance from previous reference point and change points for the word “computer”

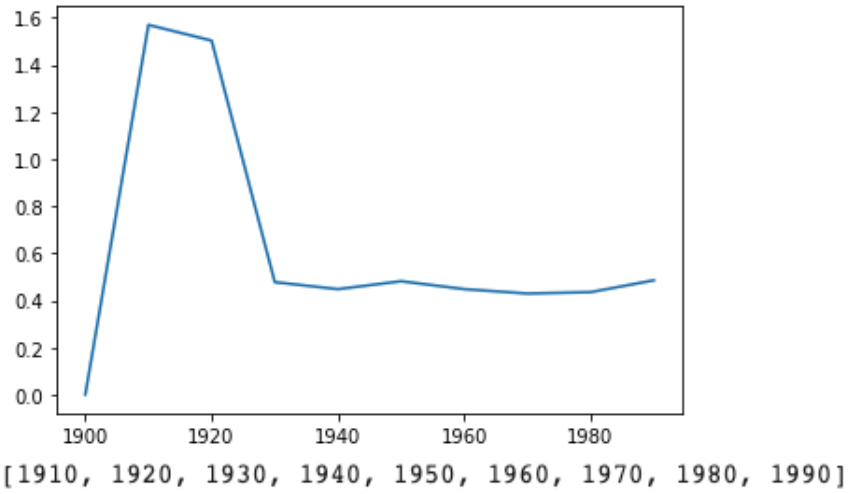

Figure 2: Distance from previous reference point and change points for the word “objectives”

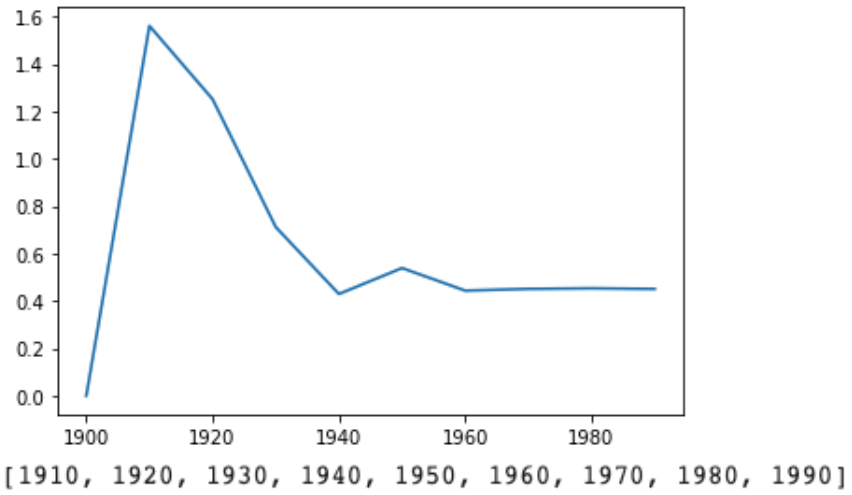

Figure 3: Distance from previous reference point and change points for the word “programs”