NLU HW1

Beisong Liu
UID: bl1986

February 25, 2025

Question 1

Part (c)

The `__getitem__` function expects the input to be an iterable list of words, so that it can find the word vectors for each word in the list. However, when the input is a single word like `"the"`, it treats this string as an iterable of characters, attempting to find the word vectors for `'t'`, `'h'`, and `'e'`. Since the dataset does not contain a word vector for `'h'`, it raises the following error:

```
KeyError: 'h'
```

Question 4

Part (a)

Embedding Space	Semantic	Syntactic	Overall
GloVe 50	40.0%	27.6%	33.2%
GloVe 100	44.5%	27.8%	35.4%
GloVe 200	31.7%	21.7%	26.2%
Skip-gram 300	50.0%	55.9%	53.3%
CBOW 300	15.5%	53.1%	36.1%

4. How do your `GloVe` results compare to the results for the two word2vec models (`Skip-Gram` and `CBOW`) reported in Table 4 of Mikolov et al. (2013)? Does the dimensionality of the embedding space have any effect on the accuracy of the analogy question?

The `skip-gram` model has better performance for all relation type. For the `CBOW` model, my `GloVe` model has better performance in the type of semantic relation, but worse performance in syntactic and overall types.

For all the relation types, a similar scenario happens: when I increase dimensionality of the embedding space from 50 to 100, the accuracy increases. But when increasing to 200, the accuracy decreases dramatically, even less than the accuracy for dimension 50. I think this is due to "The Curse of Dimensionality". When the embedding dimension is too high, words are mapped into a higher-dimensional space with a sparser representation, making the distances and similarity computations between word vectors less meaningful.

Part (b)

Embedding Space	Semantic	Syntactic	Overall
GloVe 50	56.6%	53.6%	55.0%
GloVe 100	66.5%	65.9%	66.2%
GloVe 200	70.5%	67.2%	68.7%
Skip-gram 300	50.0%	55.9%	53.3%
CBOW 300	15.5%	53.1%	36.1%

My model has a higher accuracy for all relation types compared to both `CBOW` model and `skip-gram` model.

Part (c)

Analogy Question	Gold Answer	GloVe 50	GloVe 100	GloVe 200	Skip-gram 300 (paper)
france : paris :: italy : x	rome	rome	rome	rome	rome
france : paris :: japan : x	tokyo	tokyo	tokyo	tokyo	tokyo
france : paris :: florida : x	tallahassee	miami	miami	miami	tallahassee
big : bigger :: small : x	smaller	larger	larger	smaller	larger
big : bigger :: cold : x	colder	warmer	cooler	colder	colder
big : bigger :: quick : x	quicker	quicker	quicker	quicker	quicker

Please comment on your results. How do the different embedding spaces compare to one another? How do they compare to the results reported by Mikolov et al.?

For semantic relation type (first three analogy questions), all the embedding spaces give the same answer. The first two are right, but interestingly, when trying to solve france : paris :: florida : x, all give miami as the answer, different from the right answer tallahassee.

For syntactic relation type, higher dimension embedding spaces generally give better answer, the most representative one is: warmer - cooler - colder.

When compared to the results reported by Mikolov et al., my model gets the right answers for all the questions of syntactic relation type, but the `skip-gram` model answers big : bigger :: small : x wrong. But for semantic relation type, the `skip-gram` model gets the right answers for all the questions, but my model get the france : paris :: florida : x wrong.