h1-written

kgz2437

February 2023

1 Question 1

Question 1c: The input to the function '__getitem__' is assumed to be iterable of strings. So when you pass in a single string, it can't iterate over multiple strings, so it instead iterates over each character in the string.

2 Question 2

3 Question 3

4 Question 4

4a: It seems like GLOVE embedding dimension with 100 dimensions obtain the best performance in all three categories (see Table 1). Compared to Mikolov's results, GLOVE embeddings have a higher performance on the semantic task than CBOW but not skip gram. However, both of these have a higher performance on syntactic tasks. Overall, their performance is higher than ours.

	Glove 50	Glove 100	Glove 200
semantic	0.400	0.445	0.317
syntactic	0.276	0.278	0.217
overall	0.332	0.354	0.262

Table 1: Performance of glove embeddings in three categories

4b: matching improves when we increase the leniency, by almost double (see Table 2). When we increase leniency, the performance also improves as we increase the embedding size. This might be because as we increase dimensionality, there will a larger number of close neigbhours to each vector due to the curse of dimensionality, making it more difficult to find the closest one.

4c: The embeddings spaces give the correct answer for the 1st, 2nd semantic. However, when we switch from country-capital to state-capital, the embedding space gives an incorrect answer. Additionally, larger embedding space seems to

	Glove 50	Glove 100	Glove 200
semantic	0.566	0.665	0.705
syntactic	0.536	0.659	0.672
overall	0.550	0.662	0.687

Table 2: Increasing leniency

do better on syntatci examples. However, all embeddings failed to match cold to colder and quick to quicker. (see Table 3)

	Analogy Question	Answer	Glove 50	Glove 100	Glove 200
0	france : paris : italy	rome	[rome]	[rome]	[rome]
1	france : paris : japan	tokyo	[tokyo]	[tokyo]	[tokyo]
2	france : paris : florida	tallhassee	[miami]	[florida]	[florida]
3	big : bigger : small	smaller	[larger]	[larger]	[smaller]
4	big : bigger : cold	colder	[cold]	[cold]	[cold]
5	big : bigger : quick	quicker	[quick]	[quick]	[quick]

Table 3: Qualitative Analysis of Embedding Peformance