CSU44061 Machine Learning | Weekly Assignment 9
Brendan McCann | 20332615

Part (i)

(a)

Briefly describing the following datasets:

- `input_childSpeech_trainingSet.txt`

  This dataset contains a series of short sentences with poor grammar and vocabulary - in other words, by a child. The vocabulary size is 40. The sentence length ranges from 13 to 85 characters long.

- `input_childSpeech_testSet.txt`

  This dataset contains a similar style as the previous one, except with different and unseen sentences. The vocabulary size is also 40, with a sentence length ranging between 13 and 83 characters long: very similar to the training set.

- `input_shakespeare.txt`

  The style is much different. This text is from a play, so we have character names preceding their lines - for example: `MARCIUS:....` The language is richer and more advanced. The vocabulary size is 25,671 and sentences range from lengths of 2 to 63 characters.

(b)

Seeing as the vocabulary of `input_childSpeech_trainingSet.txt` is very small and limited, I chose to reduce the number of word embeddings in the model from the default 384 to 40, which is the vocabulary size (unique characters). This reduced the model parameters to 127,320. My intuition was that it is unnecessary to have more word embeddings than the vocabulary size.

I kept the other hyper-parameters at the same values because the model parameter size dropped drastically after reducing the number of word embeddings.

(c)

(d)

There is one bias term for each of Key, Value and Query vectors.

The key vector is used to measure how relevant each element in the input sequence is to the query. The inclusion of a bias term would either increase or decrease the relevance of other tokens.

Below is a sample generation of my model with only the key bias enabled. The results were very poor and indecipherable as you can see. I believe this is because there is too much emphasis placed on the relations between the tokens as opposed to the value of the token itself. This can be addressed by modifying the bias of the value vector.

```
Dady reade read sad me de y
```

The value vector holds the actual information or meaning of the token that will be passed to the next layer. The inclusion of a bias term would reduce the importance of common words such as "the, and, etc.". In our case, this is what we want in the context of child speech - the simpler the language, the better. Even if the model fails to capture the semantics or grammar (which may be a by-product of introducing bias for value), it will likely work ok because child speech is often not grammatically correct.

Below is a sample generation of my model with only the value bias enabled. We can see the relationships between words are stronger - `Read` is clearly related to `book`, `Big` to `truck`, and so on. As a whole, the entire sentence is not very coherent. This is likely because the query bias was not set.

```
Read book Big truck Saw big flufy dont
```

The query vector represents what a token 'looks for' in other tokens. The inclusion of a bias term will impact the magnitude of the search, either making it more or less scoped. In the context of child speech, my intuition is that we would prefer to focus its scope on tokens very close to each other, as opposed to ones that are far away.

Below is a sample generation of my model with only the query bias enabled. We can see there are some spelling issues, however the grammatical and syntactic structure is quite solid.

```
More bubblas Whatt is I run fasst I hide
```

(e)

Part (ii)

(a)

(b)