Question Answering

1 CBOW Model

1.1 Model Result

After training and testing *CBOW Model*, we can get the following outputs to indicate the performance of this model:

	Loss	**Acc1**	**Acc2**	**EM**	**F1**
Validation Set	8.7346	6.494%	7.986%	5.485%	11.167%
Test Set	8.5262	5.173%	5.225%	3.104%	9.491%

From this we can see that the performance of this model is very bad, but this still in our expectation because it's a very simple model. For manual testing the example paragraph and question shown in the beginning of the assignment, we can get the answer:

chalcogen group on the periodic table and is a highly reactive nonmetal

1.2 Model Analysis

In *CBOW Model*, the vectors from the question words are averaged, with this *CBOW Model* smooth over a lot of the distributional information. Also, the *CBOW Model* is learning to predict the word by the context, or we can say, it chooses the maximal of the probability of the target word by looking at the context. So, this will happen to be a problem for rare words. Thus, the *CBOW Model* might be a useful model for small datasets with less rare words.

And, since it chooses the max possibility range of span from the context for the answer, this might be in a larger range than the actually answer. And because of EM sore require for exact match, this make the Model have very poorly EM score, while F1 score just need partially correct, which will have higher score than the EM score.

2 GRUs RNN

2.1 Models Result

The implementation of bidirectional RNN is use same hidden size from config as the unit number for both sides, so the final outputs will use the mean of the two different direction's output.

$$output = (outputs\_fw + outputs\_bw) / 2$$

After we implement the GRU with RNN, and use it instead of CBOW embeddings, we can get the performances with different dropout rate as the following:

Dropout Rate	**0.1**	**0.3**	**0.5**
Loss on Validation Set	5.879	5.827	5.885
Acc1 on Validation Set	20.67%	21.63%	20.58%
Acc2 on Validation Set	23.96%	23.69%	24.27%
EM on Validation Set	17.86%	19.31%	17.99%
F1 on Validation Set	27.58%	28.31%	27.70%

Dropout Rate	**0.1**	**0.3**	**0.5**
Loss on Test Set	6.5075	6.4778	6.5262
Acc1 on Test Set	17.460%	17.848%	17.563%
Acc2 on Test Set	20.435%	21.262%	20.072%
EM on Test Set	15.184%	16.218%	15.106%
F1 on Test Set	25.350%	27.061%	24.934%

For manual testing the example paragraph and question shown in the beginning of the assignment, we can get the answer:

- Dropout rate 0.1

the chalcogen group on

- Dropout rate 0.3

table

- Dropout rate 0.5

nonmetal and oxidizing agent that readily forms compounds (notably oxides) with most

2.2 Model Analysis

From the output of *GRU RNN Model*, we can see that the no matter which dropout rate we choose, it is always better than the *CBOW Model* - the F1 score and other measurements are way more higher than the *CBOW Model*.

And, by choosing different dropout, this have quite different results. Such as when dropout rate is 0.3, the model will have the best performance with validation set, and with manual test on the example paragraph and question, we can also say that: Although none of the model produce the right answer, model with dropout rate 0.3, will be the best one, because it return actually one word for the question, even the position of the word is still not correct, but better than other 2 models which return the long sentences.

3 Attention

3.1 Models Result

With the report from previous Models, we can now build the Attention on top of the *GRU RNN Model* with dropout rate as 0.3. And with this we can get the following performance.

Dropout Rate 0.3	**Loss**	**Acc1**	**Acc2**	**EM**	**F1**
Validation Set	5.7206	22.949%	25.757%	19.789%	30.115%
Test Set	6.3805	18.262%	21.081%	16.891%	27.187%

For manual testing the example paragraph and question shown in the beginning of the assignment, we can get the answer:

a chemical element with symbol O and atomic number 8

3.2 Model Analysis

With the output performances, we can find that when dropout rate is 0.3, even though the F1 score of the *GRU RNN model with Attention* are similar as the one without the Attention, but the manual testing of the paragraph and question, we can find that the answer finally find the right place about the question. Although this answer is not as correct as the exactly match of "8", but it's also a great gain, because our model understands the subject of the paragraph and the question is about the **Oxygen**.

And for other dropout rate we checked, we can know that it's either performance poorer (dropout 0.2) than the dropout as 0.3 version, or the manual check failed to understand the subject of the paragraph and the question.

In conclusion, we can say the *GRU RNN model with Attention* with dropout rate as 0.3 will give us the best model so far.

3.2.1 Other Dropout Models

And with other dropout rates, we can get:

Dropout Rate 0.2	**Loss**	**Acc1**	**Acc2**	**EM**	**F1**
Validation Set	6.4707	17.244%	18.912%	14.392%	21.886%
Test Set	7.0385	14.434%	16.529%	11.950%	21.316%

For manual testing the example paragraph and question shown in the beginning of the assignment, we can get the answer:

most

Dropout Rate 0.1	**Loss**	**Acc1**	**Acc2**	**EM**	**F1**
Validation Set	5.6564	23.080%	26.020%	19.789%	30.198%
Test Set	6.3410	18.960%	21.883%	16.787%	27.956%

For manual testing the example paragraph and question shown in the beginning of the assignment, we can get the answer:

that

4 Try other Models

To have the better model, I think we can add the layers to the *GRU RNN Model with Attention*. And here are the results:

4.1 2-Layers GRU RNN Model with Attention

For this model I train it with 40000 iteration, and the results are:

	Loss	**Acc1**	**Acc2**	**EM**	**F1**
Validation Set	5.5889	23.607%	26.942%	20.930%	30.611%
Test Set	6.2512	19.684%	22.452%	18.107%	28.747%

For manual testing the example paragraph and question shown in the beginning of the assignment, we can get the answer:

nonmetal and oxidizing

4.2 4-Layers GRU RNN Model with Attention

For this model I tried to train it with 80000 iteration, but since there has not enough time for it, here is the results that with only 2000 iteration:

	Loss	Acc1	Acc2	EM	F1
Validation Set	6.0226	19.614%	23.387%	16.806%	26.163%
Test Set	6.7529	14.511%	18.805%	13.528%	23.241%

For manual testing the example paragraph and question shown in the beginning of the assignment, we can get the answer:

nonmetal

4.3 Multi-Layer SOTA RNN

Because of the SOTA version of the RNN will only use the forward output in the odd layers and use backward output in the even layers. We can update our forward method to use this. Sadly, don't have enough time to experiment this model.

4.4 Conclusion

From the results we can find that the *2-Layers GRU RNN Model with Attention* will have better performance than the 1-layer version. But since the manual test didn't come out some answer that related with our question, I will not like to declare that this model is better than the 1-layer version.

Hopefully the fully trained (80000 iteration) 4-layers version can also reach an even better performance than the 2-layers version. And the Multi-Layer SOTA RNN might reach the best.