COMP5046 – Assignment 2

Johnny Peng

Jpen6856@uni.sydney.edu.au

1. Data preprocessing

NER (Named-Entity Recognition) is a N-to-N classification task, which means that we are unable to perform a lot of data preprocessing techniques such as removing punctuation, and contraction expansion, that is because these techniques will change the length of the input sentences, which needs to stay the same in order to match with the length of the output labels.

Thus, the following are all the data pre-processing techniques applied in order to the datasets:

- Tokenization on space
- Lemmatization
- Linguistic Feature extraction with Spacy

2. Input Embedding

The final input embedding has 1185 dimensions consists of the following features:

- Glove-twitter-50 (50 Dimensions)
- Part-of-Speech Tagging outputs from Spacy (48 Dimensions)
- Dependency Parsing outputs from Spacy (44 Dimensions)
- Named-Entity Recognition outputs from Spacy (19 Dimensions)
- BERT embedding (1024 Dimensions)

The feature extraction code can be found from the submitted ipynb file, except for the BERT embedding which is encoded by the pre-train "BERT-Large, Uncased" model from Bert-as-Service [1]. Please contact me if you are interested to see the code that produced the BERT embedding in this assignment.

Justification for including these features in my final input embeddings:

- Glove word embedding was included due to the following reasons:
  - It was the default input embedding in lab 9, so it would be good to include it so any additional input embedding will be more comparable to the base model.
  - Glove-50 was used in the final input embedding as it provides a significantly better accuracy over Glove-25 while switching to Glove-100 or Glove-200 does not provide any noticeable improvement in accuracy.
  - Glove word vector is often compared with word2vec and fast text, however, I didn't include word2vec and fast-text because they are predictive based word vector which is similar to BERT embedding (which I have included in the final input embedding), whereas Glove is a statistical-based word vector, so Glove word vector is more preferable over word2vec and fast-text since it will not overlap with the purpose of BERT embedding.
  - When I was training the model, I have also noticed that even if BERT embedding was included, but the accuracy will still drop significantly if I remove Glove word embedding from the final input embedding, which supports the idea of Glove embedding provides non-predictive based word vector information which is complementary to the BERT embedding.

- Part-of-Speech (PoS) Tagging output from Spacy was included in the final model due to the following reason:
  - Named-Entities would generally be classified as a proper noun, so I believe PoS tagging outputs can help with NER task in this assignment, and it does provide 0.31% improvement in accuracy based on the ablation studies.

- Dependency Parsing output from Spacy was included in the final model due to the following reason:
    - Dependency Parsing outputs were included as it might be able to provide some coreference relationship between the named-entity, and hence improve the NER model's accuracy. Although based on the ablation studies, it did not improve the accuracy, removing it from the final model will lead to a significant drop in accuracy, so it was included in the final model.
- Named-Entity Recognition outputs from Spacy was included in the final model due to the following reason:
    - By including the NER predictions from Spacy, we essentially created an ensemble NER model here, which leads to a final model with better prediction accuracy. Based on the ablation studies, it improved the model performance by 1.5%, which is a significant improvement over the base model performance, so it was included in the final model.
- BERT-encoded word embedding was included in the final model due to the following reason:
    - BERT-encoded word embedding was included because BERT is a SOTA pre-trained model with proven and robust results for improving the model performance of any NLP tasks. Based on the ablation studies, it improved the model performance by 0.74%, which is a significant improvement over the performance of the model without BERT embedding, so it was included in the final model.

Even though there are a lot more features (e.g. TF-IDF) that could potentially improve the model performance, however, due to time constraints, I have used the above features only. Ablation studies for these features were also performed, detail evaluation results can be found from the evaluation session.

3. NER model

The NER model used for this assignment is a modified version of the Bi-LSTM CRF model from lab 9. The main difference for this model is that the attention mechanism was introduced in a way shown in the graph located on the next page, detail intuitions for this attention design are also shown below.

Intuition for the attention design:

The rationale behind the attention design in this model is as simple as the following:

a. Firstly, let *h_t* represents the outputs from the last layers of the Bi-LSTM at time step t, and *h_n* represents the output from the last hidden state of the Bi-LSTM.

b. Thus, in theory, *h_n* should contain the aggregated information from all the time steps of the input sequence, and hence the attention output from *h_n* Dot-Product with *h_t* at each time step t should represent the importance of each token (with respect to the NER tasks) in the input sentence.

c. Then we concatenate the attention output with *h_t* at each time step to produce the final inputs to the dense layer and CRF.

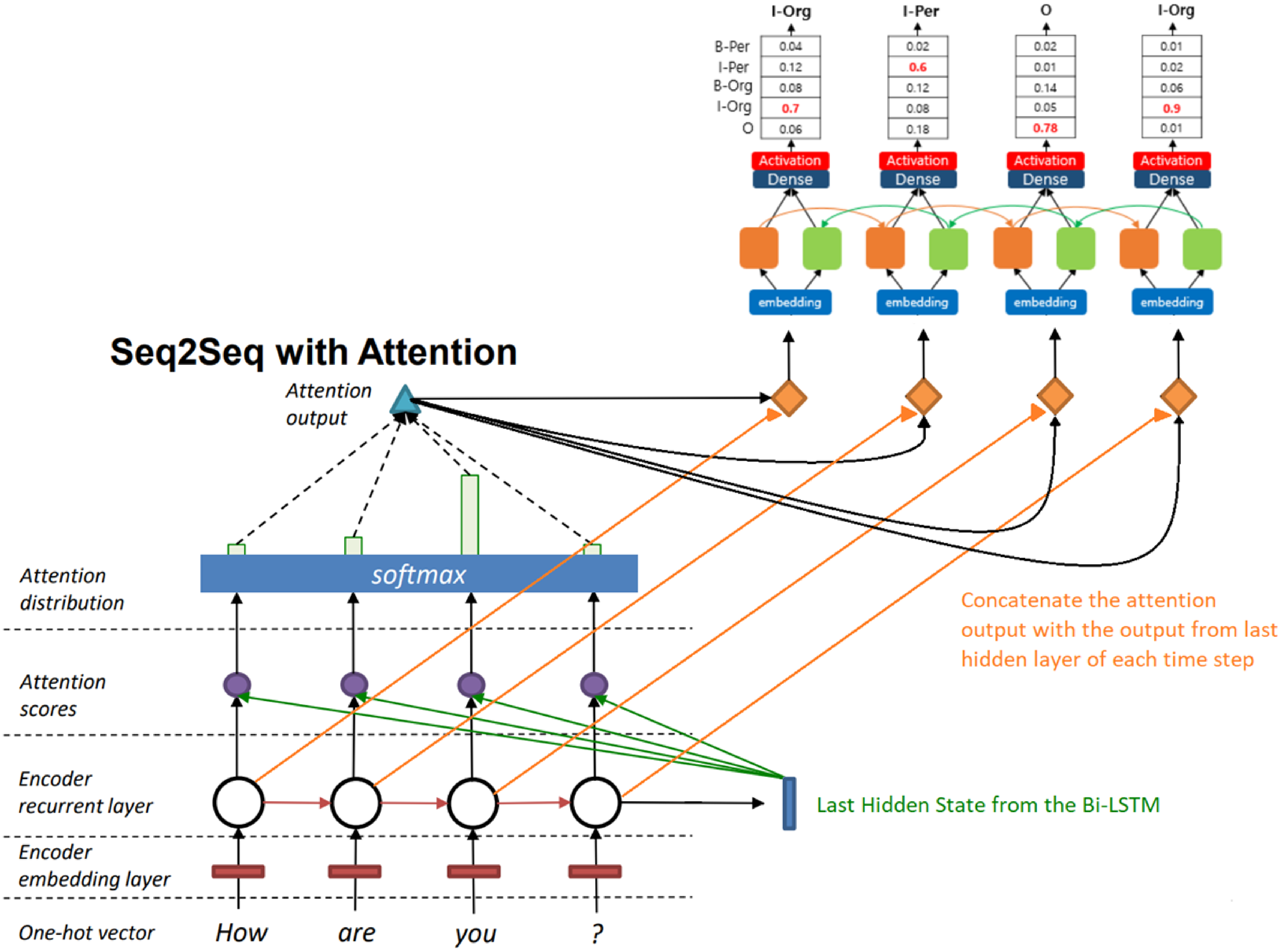
d. This concatenated input, in theory, should be superior to the raw outputs from Bi-LSTM's hidden layers, as it contains additional attention information which allows the downstream dense layers and CRF to understand about the importance of each token (with respect to the NER tasks) in the input sentence.

Based on the evaluation results in session 4b of this report, we can see that, for the same input embeddings, the introduction of attention leads to 0.67% increase in validation accuracy comparing with the base model (unmodified Bi-LSTM CRF model from lab 9). Different attention score functions were also been tested in the ablation studies. Overall, Dot-Product and Scale Dot-product score function has similar prediction accuracy, and Dot-Product score function was kept in the final model, this is purely due to it is slightly less computationally intensive than Scale Dot-product while the prediction accuracy was not compromised.

Justification for the layer strategy:

In terms of the layer strategy, the ablation studies on Single layer, 2-layers, and 3 layers Bi-LSTM were performed. Based on the evaluation results, we can conclude that 2-layers Bi-LSTM gives the best validation accuracy within 10 epochs. One possible explanation on why 3-layers design is inferior to 2-layers design in terms of validation accuracy is that 3-layers models are harder to optimize due to deeper structure and more parameters, given all the evaluations results were produced based one single hyper-parameter setting, it would be unfair to conclude 3-layers Bi-LSTM perform worse than 2-layer Bi-LSTM. However, due to time constraint, we are unable to test different hyper-parameter settings for every single model, so the 2-layer Bi-LSTM architecture was adapted in the final model.

Seq2Seq with Attention



Concatenate the attention output with the output from last hidden layer of each time step

Last Hidden State from the Bi-LSTM

4. Evaluation

a. Evaluation setup

Due to the limitation on time, all the evaluation results were trained with the following hyperparameters on Colab TPU:

- Optimizer = Adam
- Number of Hidden Units = 100
- Dropout probability = 0.5 for both LSTM and the dense prediction layer
- Learning Rate = 0.001 with weight decay of 0.0001
- Number of Epoch = 10
- Batch Size = 1 at the sentence level, in other words, each batch contains one sentence, and sequence length can be varied.

b. Evaluation result

The validation accuracy shown below represents the highest accuracy on the validation set that is produced within 10 epochs when training the model, hence they might not be the prediction results after 10 epochs of training.

Model Design	Validation Accuracy	Strategy	Included in the Final Model?
Base model (with Glove-50)	93.44%	-	Yes
+ Named-Entity Embedding	94.93%	Additional Input Embedding	Yes
+ Pos Tag Embedding	95.24%	Additional Input Embedding	Yes
+ Dependency Embedding	95.02%	Additional Input Embedding	Yes
+ Attention (Scaled Dot-Product)	95.91%	Attention Strategy	No
= Attention (Dot-Product)	95.91%	Attention Strategy	Yes
+ BERT Embedding	96.65%	Additional Input Embedding	Yes
+ 1 more Bi-LSTM layer (2 layers)	97.45%	Layer strategy	Yes
+ 1 more Bi-LSTM layer (3 layers)	96.15%	Layer strategy	No

Note that, the ablation studies were conducted in the order of top-to-bottom as shown in the table above, it’s possible that the evaluation results will differ or different conclusion will be made if the ablation studies were done in a different order, however, due to limitation on time, the ablation study above is the only evaluation results that we got.

The “+” sign in the “Model Design” column represents this is an additional feature to the previous model design (the model shown one row above), while “=” sign represents this feature replaced the equivalent feature in the previous model design.

References

1. Bert-as-Service GitHub page, https://github.com/hanxiao/bert-as-service, last accessed [2020/06/05].
2. Some of the codes, graphs are modified based on the lecture notes and codes from lab 9, the exact parts that was referenced will be provided upon request.