Improvement on Knowledge-backed Generation Model Using Post-Modifier Dataset (PoMo)

Priyanka Nath / SBU ID: 112715634
Shanuj Shekhar / SBU ID: 112670784
Trisha Kanji / SBU ID: 112686934

Abstract

Post-modifier is a short descriptive phrase that comes after a word (usually a noun) in a sentence which gives more detailed contextual information about that word. Post-modifier generation(PoMo Generation) is basically a contextual data-to-text generation problem, where a good post-modifier generation model selects relevant facts about the word, given the text and outputs a post-modifier covering these facts in such a way that it maintains the context with respect to the original text. In this project, we aim to replicate such a PoMo Generation model [1], and try improving its accuracy.

1 Introduction

The main objective is to improve the already existing knowledge-backed generation model which uses the PoMo data-set[1]. Precisely, the task of post modifier generation requires to automatically generate a post modifier phrase describing the target entity (an entity essentially refers to a noun but here we only consider people) that contextually fits in the input sentence. One can always look up Wikipedia for information concerning a particular topic but it is always convenient to get relevant and concise information instead of being bombarded with tons of irrelevant information. It is an interesting problem to train a model that understands the context pertaining to. In the journalism domain post-modifier generation can be viewed as an assistive writing task. News articles are adorned with post modifiers to give the readers brief contextual information about a particular entity.

Broadly speaking a successful post modifier generation model has two sub tasks - selecting a set of facts or claims about the concerned entity and from that set selecting the most relevant fact to generate a post modifier that contextually describes it. [1]

The paper we referred for the project concluded that PoMo generation models perform well when the claims are given but they suffer miserably if relevant claims are not given and have to be selected by some claim selection method. This is because post modifier can be ambiguious. Moreover since we are using the previous and next sentences as context for a given entity, any contextual information that is not explicitly mentioned will not be understood implicitly. Examples are further explained in Section 2.2.

Possible reasons for the bad performance could be that claim selection is not being performed effectively. This is supported by the fact that the baseline models perform well when supplied with the relevant claims. Hence claim selection should be improved for optimal performance. Another way to improve performance of the PoMo generation model would be to modify the generative model itself.

The ideas were tested by tuning the baseline model hyper-parameters and experimenting with different model architectures.

The changes were evaluated in terms of prediction accuracy and prediction score. We tried and tested our generative model with different global attention functions and different combinations of hyper-parameters. In order to optimize claim selection we aimed to different activation functions which were evaluated in the same manner.

We trained the different neural models for 100 and 800 epochs respectively, and the results for both the cases deemed the model using bi-Directional LSTM encoder/decoder module to be the best performing one, in terms of Prediction Score. Thus, for this model we try to improve the PoMo by altering the attention mechanism for the model. The Global attention function (Luong) gave us the optimal results.

2 Your Task

Our task can be broken down into two subtasks namely improving claim selection and improving PoMo generation. Figure 3. shows the basic overview of our task. Claim selection model takes all the entity claims as input and outputs relevant claims depending on how it is implemented. Since we use soft claim selection, the claims are selected during the PoMo generation. For claim selection we can choose a threshold value which can be used to classify a fact as relevant if it score above the threshold or we can use a ranking based model which outputs the top 'n' number of relevant claims in terms of their scores. These relevant claims are fed as input along with the context sentence embeddings using which the PoMo generation model predicts the most likely post modifier.

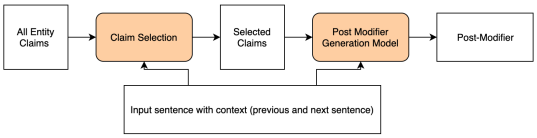

Figure 1: General PoMo Generation Model with soft claim selection

2.1 Baseline Model(s)

For PoMo Generation we used a seq2seq model with attention which utilized OpenNMT. The model as shown in Figure 2. is a 2-layer biLSTM that encodes the concatenation of the sentence and claim embeddings and outputs a post modifier.

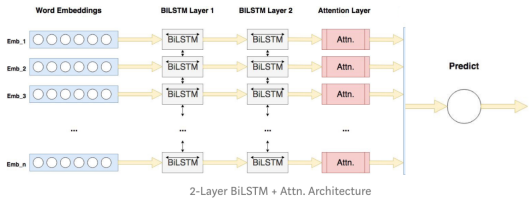

Figure 2: 2-layer biLSTM Model Architecture with attention

2.2 The Issues

One big issue with implementing this project is the time taken for training. The baseline provided was trained for 1,000,000 epochs. We trained for 100 and 800 epochs which took us approximately 1.5 hours and 8 hours respectively. This along with limited computational power proved to be a very stressful hurdle.

An actual issue that was identified was in the implementation of claim selection. In Figure 3. the first sentence shows that the model does well but the lack of information with respect to time causes it to discard the word 'former' which is an important modifier.

Input	Sky News reported Thursday night that Kenneth Clarke , _____, had not yet decided whether to support Mr. Howard 's candidacy , raising the possibility the party could face a divisive battle for leadership .
Claims	+ (position held: *Chancellor of the Exchequer*) (position held: *Secretary of State for the Home Department*)
Target	a former chancellor of the exchequer
All Claims	the Home Secretary
Oracle	the Chancellor of the Exchequer
Input	" A lot of people think it 's something we just started , but we actually opened the season with our first drive using it against Indianapolis , " said Howard Ballard , _____.
Claims	+ (member of sports team: *Buffalo Bills*) + (position played on team / speciality: *offensive tackle*) (mass: *325 pound*) (height: *78 inch*)
Target	Buffalo 's robust , 6-foot-6-inch , 325-pound right tackle
All Claims & Oracle	the Bills ' offensive tackle

Figure 3: Examples of sentences where issues arises

The second example shows that the model fails to score the most relevant claim the highest. Even though there exists a claim which is highly similar to the target, the model scores the other claim better which is not ideal.

3 Your Approach

We have used different neural models in the above architecture for generating post modifiers. Till now we have trained these models for 100 iterations which yielded a low accuracy, and then for 800 iterations with higher accuracy. This is because of the computational limitations of our systems.

Also, in our training data-set we are given a relevance score of the post modifier with respect to its sentence. We have done claim selection using the claims, next and previous sentences which acts as contextual information. With the help of this context, we have calculated the relevance score while we are training our model.

Experimentation with Model Architecture: We have experimented with different types of neural models, namely, LSTM and GRU with Bi-directional Recurrent Neural Network, Recurrent Neural Network and Mean Encoders. Each encoder gave different accuracies and prediction score on the basis of the encoding scheme they follow.

Experimentation with Attention Functions: Our neural model employs the attention function

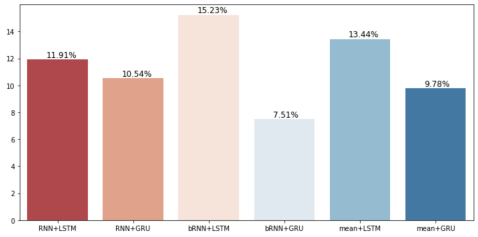

Figure 4: Accuracy of different models trained for 100 epochs. We can see that bRNN+LSTM model performs the best.

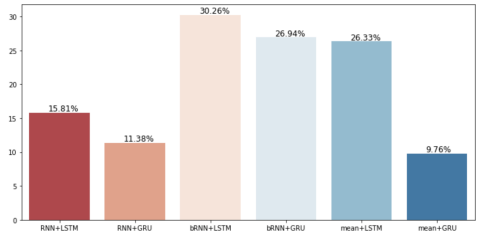

Figure 5: Accuracy of different models trained for 800 epochs. In this case as well bRNN+LSTM i.e. the baseline model has the highest accuracy of of 30.26 percent.

after the sentence has been represented using the second layer of LSTM. This attention mechanism helps to determine the relevant claim (soft claim selection) by assigning a claim level score. Each time the model predicts an output word, it only uses parts of an input where the most relevant information is concentrated instead of an entire sentence. In other words, it only pays attention to some input words. We have used 3 different global attention functions to obtain optimal results of post modifier generation. The idea of a global attention (Luong) is to use all the hidden states of the encoder when computing each context vector. The downside of a global attention model is that it has to attend to all words on the source side for each target word, which is computationally costly.

Experimentation with Hyper-Parameters: Apart from attention functions, experimenting with hyper-parameters could play a vital role in improving performance. Hyper-parameters such as epochs, learning rate, number of sentences per epoch (batchSize), etc. have been used to acquire better accuracy. Due to the excessively large data-set, we could only try for less number of epochs since we had computational limitations. So we tried for 100 and 800 epochs.

4 Evaluation

4.1 Data-set Details

PoMo dataset is created automatically from news articles. While composing news articles journalists feel a need to incorporate entity information relevant to that particular news event. Now, for this purpose facts related to that particular entity are extracted from an external data source. Here, the claims corresponding to the particular entity are extracted from wikidata.

For the purpose of creating training data, a large number of such news articles are collected and by removing the pieces of text we can obtain the data-set. Post Modifiers from the sentences in this data can be extracted by finding noun phrases(NPs) sharing relation with an entity(for ease, assumed to be people) in sentence.

PoMo consists of around 231,000 sentences with post modifiers and associated claims extracted from wiki data for around 25,000 unique entities stored in key-value pairs.

Given a file of sentences about entities with their post modifiers and a file containing claims (facts) about these entities, we have prepared the training data. This training data consists of sentences with their corresponding claims as features and post modifiers as labels.

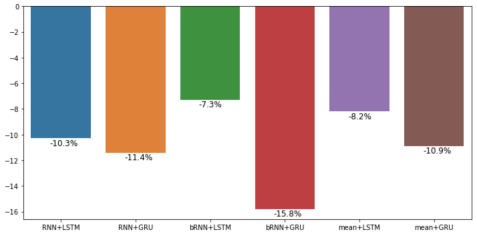

Figure 6: Prediction score for different models trained for 100 epochs. We can see has the score which is closest to 0.

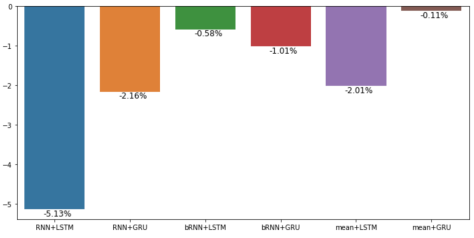

Figure 7: Prediction score for 6 different models trained for 800 epochs. In this case as well bRNN+LSTM i.e. the baseline model has best score.

4.2 Evaluation Measures

The evaluation measures how well the generated post-modifier matches the original post-modifier and whether the generated text fits with rest of the sentence. We use 2 metrics to evaluate our model - accuracy and prediction score.

4.3 Baselines

Our baseline model is generated with the help of hyper-parameters tuning. Number of instances (batchSize) is taken to be 32 as default. Optimal results were obtained when we took batchSize to be 50. Initially training of 100 epochs gave us a low accuracy since the data-set might have been under-fitted. Since increasing number of epochs would be computationally heavy so we tried for 800 epochs which took almost 12 hours to compute. Accuracy obtained at the end of 800 epochs were significantly larger compared to model trained after 100 epochs. Optimal results were found when the learning rate came out to be 1.0 for our BiLSTM encoder-decoder with attention model. This model has been trained for 800 epochs to get the optimal results.

4.4 Results

The model architecture using the 2-layer BiLSTM attention configuration gives us the best results in comparison to other RNN models experimented on (biRNN+GRU, RNN+GRU, RNN+LSTM, Mean+GRU, Mean+LSTM).

The results obtained were found to be similar in trend to baseline model.

4.5 Analysis

The attention function is the key component responsible for soft claim selection and determines which of the claims are more relevant and passes them forward by assigning the claim tokens a claim score. Thus, we experimented by varying the attention mechanisms to see if it reflects any significant change in the performance of the models and look out for any possible improvements. We tried out 3 different global attention functions namely dot product, Luong and Bahdanau attentions for the best performing model i.e. bRNN+LSTM which was the baseline model. Luong attention worked best.

While the above results present an interesting insight, it should be noted that the above results are obtained by running on a relatively smaller number of epochs and we do not yet know whether a similar trend would follow if tested on the dataset with a much greater epoch value (like 1,000,000).

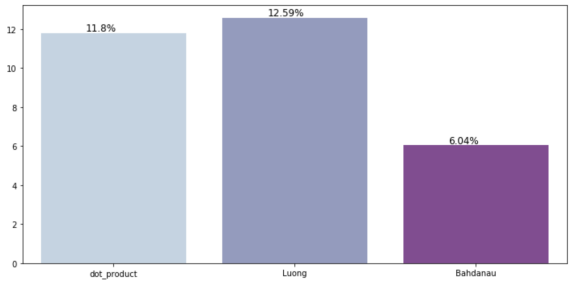

Figure 8: Accuracies for different global attention functions using the bRNN+LSTM model. Luong performs the best.

4.6 Code

The code is available at the following location - Google Drive Link More details about the code is written in the README.md file submitted along with this report.

5 Conclusions

In this project, we attempt to improve the already existing Post Modifier Generation model by tuning the hyper-parameters and experimenting with different model architectures.We realised that PoMo generation depends mainly on the context and claim selection task. Here, we mainly have to concentrate on the claim selection part, the claims would differ for different contexts. This is the reason which makes claim selection a hard problem. We verified that the baseline performs the best for 100 and 800 epochs. We tried different global attention functions on the baseline model and concluded that Luong attention works the best for our use case. Time was a big constraint for our project. The original paper showed results after training for 1,000,000 epochs but due to time and computational power limitations, we were able to verify the trend of results as mentioned in the paper.

6 References

[1] Kang, Jun Seok, I. V. Logan, L. Robert, Zewei Chu, Yang Chen, Dheeru Dua, Kevin Gimpel, Sameer Singh, and Niranjan Balasubramanian. "PoMo: Generating Entity-Specific Post-Modifiers in Context." arXiv preprint arXiv:1904.03111 (2019).

[2] Puduppully, Ratish, Li Dong, and Mirella

Lapata. ”Data-to-text generation with content selection and planning.” In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6908-6915. 2019.

[3] Nie, Yixin, Haonan Chen, and Mohit Bansal. ”Combining fact extraction and verification with neural semantic matching networks.” In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6859-6866. 2019.

[4] https://medium.com/@joealato/attention-in-nlp-734c6fa9d983