since 1995

Data Science Natural Language Processing NLP Papers Summary

Day 206: NLP Papers Summary – Transformers And Pointer-Generator Networks For Abstractive Summarization

By Ryan 24th July 2020 No Comments

Objective and Contribution

Explored different mechanisms on the transformer model and assess the impact it has on abstractive summarisation. The mechanisms include n-gram blocking, coverage loss, and pointer-generator (PG) network. The mechanisms attempt to alleviate the repetition and factual inconsistent problem. Results show that ROUGE scores improved as a result of these mechanisms.

Methodology

Our baseline model is the transformer model. We explored two techniques to counter repetition problem:

1. N-gram blocking
2. Coverage loss

WHAT IS N-GRAM BLOCKING?

N-gram blocking is added to our decoder. Decoder uses beam search to construct summaries and as it is selecting the next word, n-gram blocking would eliminates those words that would lead to an n-gram that already exists within the beam.

WHAT IS COVERAGE LOSS?

The coverage loss involves creating a coverage vector that's the sum of attention over all previous time steps. With the coverage vector, coverage loss is compute by minimising between the attention vector and the coverage vector.

PG NETWORK TRANSFORMER

With PG Network, our transformer has the ability to copy or generate words at each given time step. The generation probability is computed using the hidden state, context vector, and decoder input.

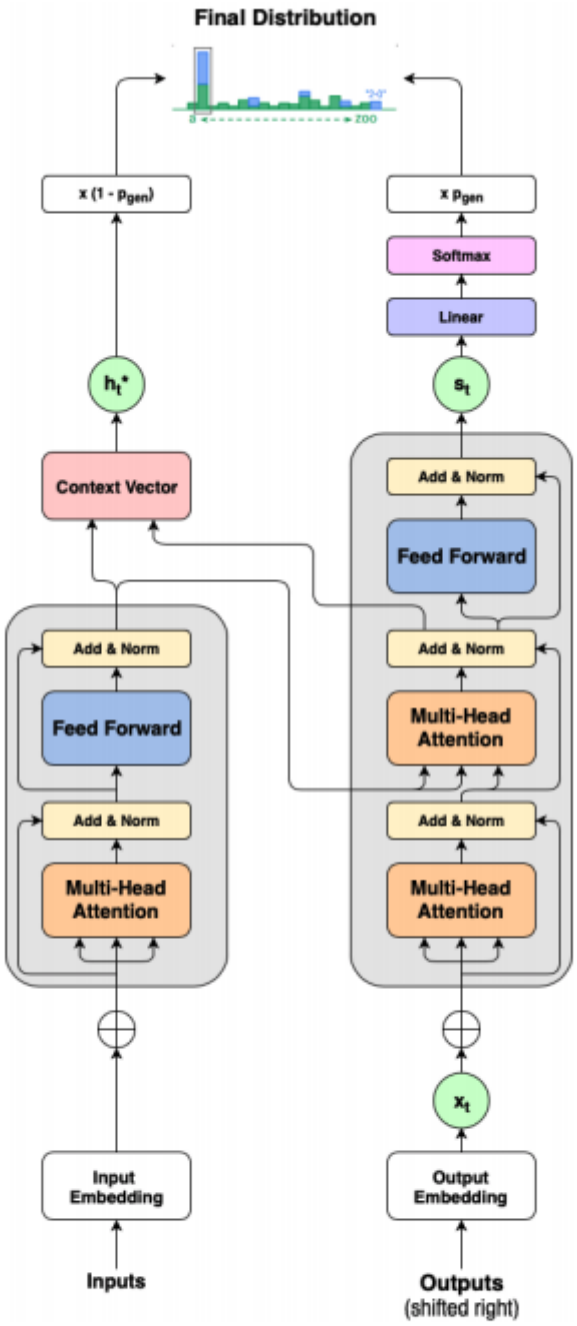

Figure 1: Pointer-generator network on a transformer; marks s_t, h_t^* and x_t for calculating p_{gen}. This original diagram is based on the diagram of a transformer presented in *Attention is all you need*[19].

Experiments and Results

The evaluation dataset is CNN/DailyMail and the evaluation metric is the ROUGE scores. We ran different variants of our PG-Transformer:

1. PG-Transformer
2. PG-Transformer + Coverage
3. PG-Transformer + N-gram blocking

RESULTS

The results are displayed below. Note that the results are generated from only 24 hours of training and this is significant less than PG network, which took 4+ days and Sanjabi's transformer, which took 2+ days. Comparing the transformer baseline with our PG-transformer, we can see that PG network was able to improve the performance of our model. We further improved the performance of our PG-transformer by adding coverage and n-gram blocking mechanism. We found that n-gram block yielded a much larger performance increase than coverage loss.

Table 1: Results

Method	R-1	R-2	R-L
Vanilla RNN (See et al., 2017)	30.49	11.17	28.08
Pointer-Generator (See et al., 2017)	36.44	15.66	33.42
Pointer-Generator + Coverage (See et al., 2017)	**39.53**	17.28	**36.38**
Transformer (Sanjabi)	27.4	-	-
Transformer Baseline	20.23	3.45	13.43
Transformer + Pointer-Generator	22.10	4.03	14.66
Transformer + Pointer-Generator + Coverage	22.93	4.08	15.08
Transformer + Pointer-Generator + N-Gram Blocking (2-gram)	25.31	4.16	15.99

Baseline transformer generated summaries with lots of repetition and unable to handle OOV words as shown in the qualitative analysis below. Our PG-transformer was able to reduce OOV mishandling but still suffer from factual inconsistency. With the added coverage, we see that the summaries don't have repetition problem anymore, however, repetition starts to appear in ideas and phrases level. The n-gram blocking eliminates all repetitions.

Conclusion and Future Work

Future work should involve hyper-parameter tuning and extended training time to make results more comparable to SOTA. We found that n-gram blocking was more effective in reducing repetition than coverage loss and although PG network alleviated the out-of-vocab (OOV) problem, models still produce factually inconsistent summary.

Source:
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15784595.pdf

Ryan

Data Scientist

Previous Post Next Post

Day 205: Learn NLP With Me – Zero-Shot Learning for Text Classification

Day 207: Learning PyTorch - Fine Tuning BERT for Sentiment Analysis (Part One)

Data Science Natural Language Processing NLP Papers Summary

Day 365: NLP Papers Summary – A Survey on Knowledge Graph Embedding

Ryan
30th December 2020

Data Science Implementation Natural Language Processing

Day 364: Ryan's PhD Journey – OpenKE-PyTorch Library Analysis + code snippets for 11 KE models

Ryan
29th December 2020