Pointer-Generator Network Applied to Scientific Articles Summarization

Tung-Duong Mai, Martin Elingaard, Dongjoo Kim
KAIST School of Computing, DTU Compute

Background

- The need to summarize the text is growing with the increasing number of written materials.
- Two main approaches in summarization are extractive and abstractive. Abstractive summarization is harder but closer to human-writing.
- We reproduce the abstractive model in *"Get To The Point: Summarization With Pointer-Generator Networks" (See et. al)*, which deals with OOVs, factual details and repetition in vanilla seq2seq model.

Contribution

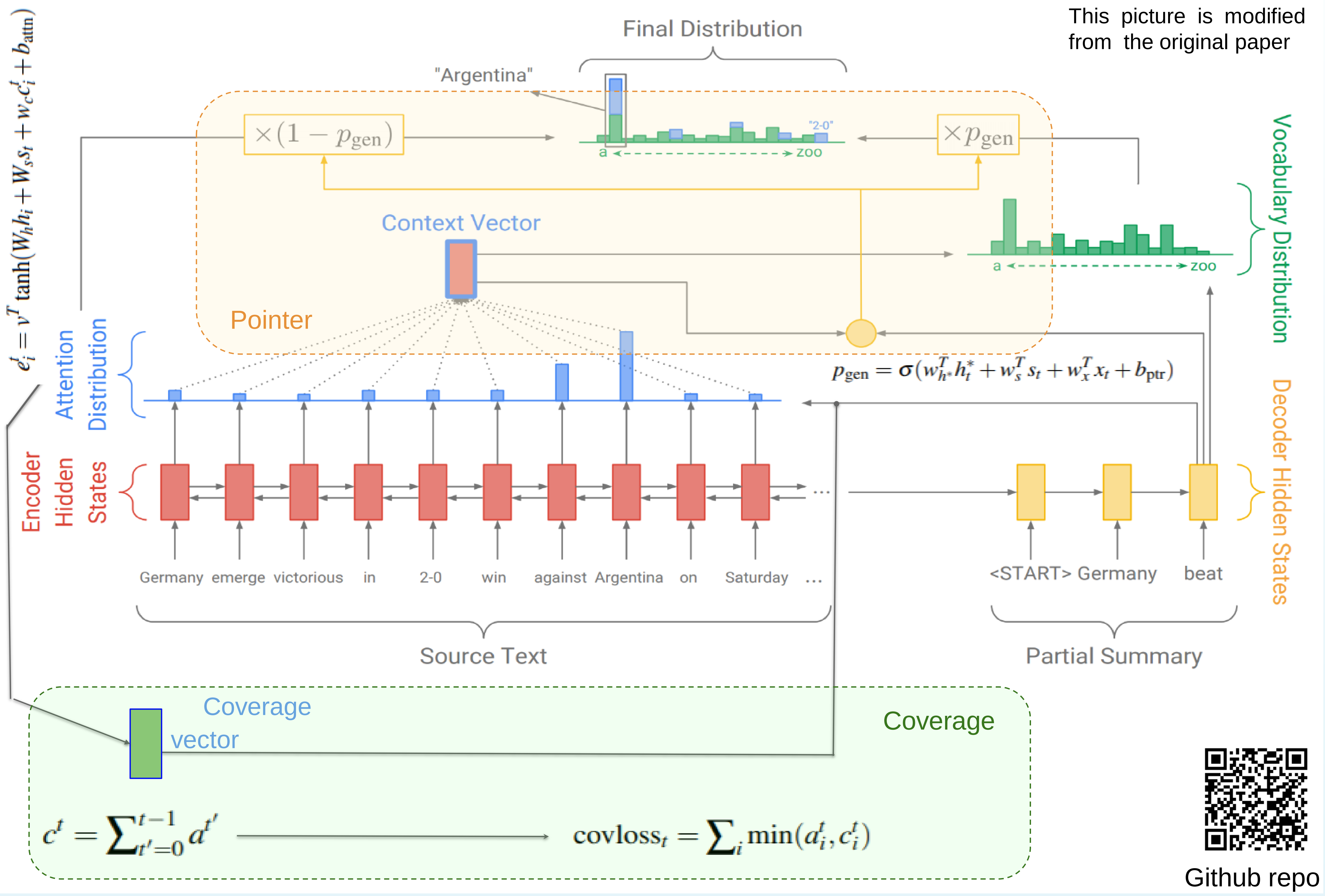
- Successfully reproduce the original paper with concise code and publish on Github.
- Change the dataset from CNN/DailyMail news to scientific articles.
- Test the model on various architectures, hyperparameters and word embeddings.
- Implement the evaluation: ROUGE metric.
- Solve the NaN problem.
- Produce meaningful result with limited computational resources.

Dataset

- Use MEDLINE dataset which comes from U.S. National Library of Medicine.
- Process it to title-gen dataset as introduced in Nikolov et al., 2018.
- Below table represents statistics (mean and standard deviation) of the dataset.

title-gen	*Abstract*	*Title*
Token count	245 ± 54	15 ± 4
Sentence count	14 ± 4	1
Sent. token count	26 ± 14	-
Overlap	$73\% \pm 18\%$	
Repeat	$44\% \pm 11\%$	-
Size (tr/val/test)	$5'000'000/6844/6935$	

Model



This picture is modified from the original paper

Github repo

Models Selection and Hyperparameters Tuning

RNN cell: **GRU** is faster and reduces the loss faster (tested on vanilla model).

RNN cell	LSTM	GRU
Loss after 10 epochs	4.48	3.11
Average time per epochs (s)	115.20	81.77

Embedding: Training from scratch using **Keras embedding** works comparably well compared to pretrained embedding but much faster.

Pretrained embedding	Keras	W2V	GloVE
Loss after 10 epochs	1.82	1.63	1.56
Average time per epochs (s)	91.44	165.67	162.80

Hyperparameters:

- Learning rate: 0.01 does not reduce the loss, 0.0001 works too slow, **0.001** is optimal

Learning rate	0.01	0.001	0.0001
Loss after 10 epochs	5.38	3.11	4.66
Average time per epochs (s)	127.58	124.78	123.28

- Unless specified by the embedding, the embedding dimensions is **128** (as the paper).
- For other hyperparameters (hidden dimensions, coverage loss normalization etc.) we follow the paper's selection. Gradient clipping and smoothing terms were used to solve NaN problem.

Results

Red: generated, Blue: target

Fabricated: *"postmortem contrast medium produce the lambert <UNK> of rat mitochondria"*
"electrophysiological study of the denervated orbicularis <UNK> muscle in dogs"

Same context: *"a novel receptor for screen for agonist inducible annexin a1 receptor for inducible cell fusion"*
"analysis of snare mediated membrane fusion using an enzymatic cell fusion assay"

Model	R-1	R-2	R-L	Token count
lead-1	0.316	0.136	0.310	28 ± 14
seq2seq - lstm ◁	0.375	0.173	0.329	12 ± 3
char2char - lstm ◁	**0.479**	0.265	**0.418**	14 ± 4
subword - conv ◁	0.463	**0.277**	0.412	14 ± 7
seq2seq - gru (ours) ○	0.331	0.044	0.363	16 ± 4
seq2seq + cov - gru (ours) ○	0.336	0.042	0.367	16 ± 4
point-gen (ours) ◇	0.351	0.036	**0.371**	15 ± 4
point-gen + cov (ours) ◇	**0.354**	**0.044**	0.368	16 ± 4
point-gen + cov (training data) ◇	0.624	0.377	0.632	15 ± 4

First three; Nikolov et al. 2018. Next four; replication of See et al. 2017 and applied to scientific articles dataset. Last; result on training data. Bold font marks the best performing model within each metric.

◁ Trained on full 5M articles dataset.
○ Trained on 70K subset of full dataset.
◇ Trained on 7K validation dataset (validated on testset)

Conclusion

A seq2seq+attention and a pointer-generator model has been applied to the scientific articles dataset. The implementation proved more challenging than first anticipated, especially due to the large size of the dataset (5M articles). Our models were therefore applied to a heavily reduced version of the dataset (≈1.5%), and show signs of overfitting. Given additional time and computational resources we are confident that our models could surpass the results produced by Nikolov et al. 2018 on the full dataset. However, with the current resources our results do no match those produced on the full dataset.

Acknowledgements

[1] Get To The Point: Summarization with Pointer-Generator Networks, See et al. 17
[2] Data-driven Summarization of Scientific Articles, Nikolov et al. 18
[3] Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond, Nallapati et al. 16