

Data Science

Natural Language Processing

NLP Papers Summary

Day 185: NLP Papers Summary – A Discourse- Aware Attention Model For Abstractive Summarization Of Long Documents

By Ryan 3rd July 2020 No Comments

Objective and Contribution

The first paper to perform abstractive summarisation on long-form documents (research papers). The architecture consists of a hierarchical encoder that captures the discourse structure of a research paper and an attentive discourse-aware decoder to generate the summary.



The contributions of this paper are:

1. Proposed an abstractive model for summarising research papers
2. Introduced two large-scale datasets of long structured research papers obtained from arXiv and PubMed

Discourse-aware Summarisation Model

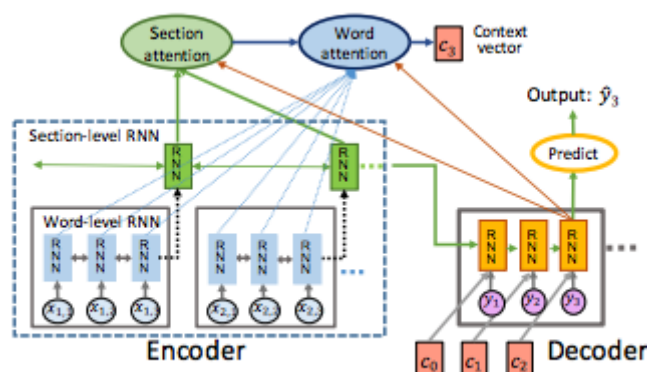



Figure 1: Overview of our model. The word-level RNN is shown in blue and section-level RNN is shown in green. The decoder also consists of an RNN (orange) and a “predict” network for generating the summary. At each decoding time step t (here $t=3$ is shown), the decoder forms a context vector c_t which encodes the relevant source context (c_0 is initialized as a zero vector). Then the section and word attention weights are respectively computed using the green “section attention” and the blue “word attention” blocks. The context vector is used as another input to the decoder RNN and as an input to the “predict” network which outputs the next word using a joint pointer-generator network.

HIERARCHICAL ENCODER

Our encoder is a hierarchical RNN that captures document discourse structure. The encoder first encode each discourse section by parsing in all the words into their respective section RNN. We then takes the outputs of all section RNNs and feed the hidden states into another RNN to encode the whole document.

DISCOURSE-AWARE DECODER

At each decoding step, our decoder takes in the words of the document and also attend to the relevant discourse section. We would use the discourse-related information to modify word-level attention function. At each decoding step, the decoder would use the decoder  and context vector to predict the next word in the summary.

COPY MECHANISM

We added an additional binary variable to the decoder to determine whether the decoder should generate a word or copy a word from the source. The copy probability is learned and optimised during training.

COVERAGE MECHANISM

We track attention coverage to avoid the problem of generating repeated phrases or words. The coverage vector includes information about the attended document discourse sections and it's incorporated into the attention function.

ArXiv and PubMed

We introduced the two research papers dataset: arXiv and PubMed. During our data collection process, we removed any documents that are too long or too short or do not have an abstract or discourse structure. We remove any figures and tables and normalise any math formulas and citation markers with special tokens. The abstract is the ground-truth. The dataset statistics are displayed below with average document length being 3000 – 5000 words.

Datasets	# docs	avg. doc. length (words)	avg. summary length (words)
CNN	92K	656	43
Daily Mail	219K	693	52
NY Times	655K	530	38
PubMed (this work)	133K	3016	203
arXiv (this work)	215K	4938	220

Table 1: Statistics of our arXiv and PubMed datasets compared with existing large-scale summarization corpora, CNN and Daily Mail (Nallapati et al., 2016) and NY Times (Paulus et al., 2017).

Experiments and Results

We used ROUGE score as evaluation metric and we compare our method with different benchmark models as below:

1. LexRank, SumBasic, LSA (extractive)
2. Attention seq2seq, PG network (abstractive)

RESULTS

The two tables below showcased the results on arXiv and PubMed dataset. The results show that our discourse-aware model was able to outperform all the baseline models, both extractive and abstractive.

Summarizer		RG-1	RG-2	RG-3	RG-L
Extractive	SumBasic	29.47	6.95	2.36	26.30
	LexRank	33.85	10.73	4.54	28.99
	LSA	29.91	7.42	3.12	25.67
Abstractive	Attn-Seq2Seq	29.30	6.00	1.77	25.56
	Pntr-Gen-Seq2Seq	32.06	9.04	2.15	25.16
	This work	$\uparrow\pm$ 35.80	$\uparrow\pm$ 11.05	\uparrow 3.62	$\uparrow\pm$ 31.80

Table 2: Results on the arXiv dataset, RG: ROUGE. For our method \uparrow (\pm) shows statistically significant improvement with $p < 0.05$ over other abstractive methods (all other methods).

Summarizer		RG-1	RG-2	RG-3	RG-L
Extractive	SumBasic	37.15	11.36	5.42	33.43
	LexRank	39.19	13.89	7.27	34.59
	LSA	33.89	9.93	5.04	29.70
Abstractive	Attn-Seq2Seq	31.55	8.52	7.05	27.38
	Pntr-Gen-Seq2Seq	35.86	10.22	7.60	29.69
	This work	\uparrow 38.93	$\uparrow\pm$ 15.37	$\uparrow\pm$ 9.97	$\uparrow\pm$ 35.21

Table 3: Results on PubMed dataset, RG:ROUGE. For our method, \uparrow (\pm) shows statistically significant improvement with $p < 0.05$ over abstractive methods (all other methods).

We also performed some qualitative evaluation and we observed that our model was able to generate summaries that not only capture the problem introduction like other SOTA benchmark models but also able to capture the methodology and impacts of the paper.





Ryan

Data Scientist

Previous Post

Day 184: Learning
<PyTorch - Machine
Translation with
TorchText

Next Post

Day 186: NLP
Papers Summary -
Contextualizing
Citations for
Scientific
Summarization
using Word
Embeddings and
Domain
Knowledge





[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

