Pointer-Generator Networks Applied to Scientific Article Summarization

Martin Ohrt Elingaard
Technical University of Denmark
DTU Compute
Kgs. Lyngby, Denmark
m.elingaard@gmail.com

Tung-Duong Mai
KAIST
School of Computing
Daejeon, South Korea
john_mai_2605
@kaist.ac.kr

Dongjoo Kim
KAIST
School of Computing
Daejeon, South Korea
wnehdrla@kaist.ac.kr

Abstract

Although advancement in neural sequence-to-sequence model had induced a feasible method for *abstractive summarization*; the model usually suffers from three issues: out-of-vocabulary, inaccurate factual details and repetitions. In this paper, we discuss two architectures that can be applied on top of the standard sequence-to-sequence attentional model: a *pointer-generator* network to copy words from the source via *pointer* and generate novel words via *generator*, and a *coverage* mechanism to keep track of summarized part and penalize repetition. By applying the model to the scientific articles *MEDLINE dataset*, we show the effect of the two mechanisms on the generated summary. We also present an analysis of the nature of *scientific articles* deduced from the summary result as well as the *ROUGE metric* in summarization task.

1 Introduction

Auto-summarization is drawing a lot of attention from researchers due to the increasing amount of written materials in the world. There are two methods in summarization: *extractive* and *abstractive*. *Extractive method* produces summary by extracting some parts of the source document. On the other hand, *abstractive models* may generate novel words not featured in the original text - resembling human-written summary. The extractive method is simple and ensures the base level of correction for the information. However, it is unable to deal with high-quality summarization features such as paraphrasing, generalization or domain knowledge, which can be solved by abstractive method.

Due to the difficulty of the abstract summarization, the most popular methods before 2014 are extractive method (Kupiec et al., 1995; Paice, 1990; Saggion and Poibeau, 2013). However, the advent of *sequence-to-sequence* model (Ilya Sutskever and Le, 2014) provide a feasible architecture to make abstractive summarization (Sumit Chopra and Rush, 2016; Nallapati et al., 2016; Alexander M Rush and Weston, 2015; Wenyuan Zeng and Urtasun, 2016) Nevertheless, these systems still shows undesirable behaviors: producing factual errors, inability to handle out-of-vocabulary (OOV) words, and repeating some phrases.

These problems are solved by the model proposed in (See et al., 2017) by using *pointer-gernerator network* with *coverage*. In (See et al., 2017), the experiment is proceeded on CNN/DailyMail news articles and produces state-of-the-art result in the domain of abstractive summarization by at least 2 ROUGE points. We will test the model with new dataset: scientific articles. We believe that summarization on scientific articles is much more challenging since these articles are very compact with a lot of unique information, scientific words, and even newly defined words in the article. While the main content of a news article is usually captured in first few sentences, it is not the case for the scientific articles where the authors usually write about the research background first. This difference in the nature of scientific articles make summarization much more challenging. We want to figure out if it also make meaningful result in this new dataset.

2 Method

Overall structure of the model

We use the model proposed in (See et al., 2017). The model is based on the (baseline) attentional sequence-to-sequence model (2.1), augmented by two architectures: pointer-generator network (2.2),

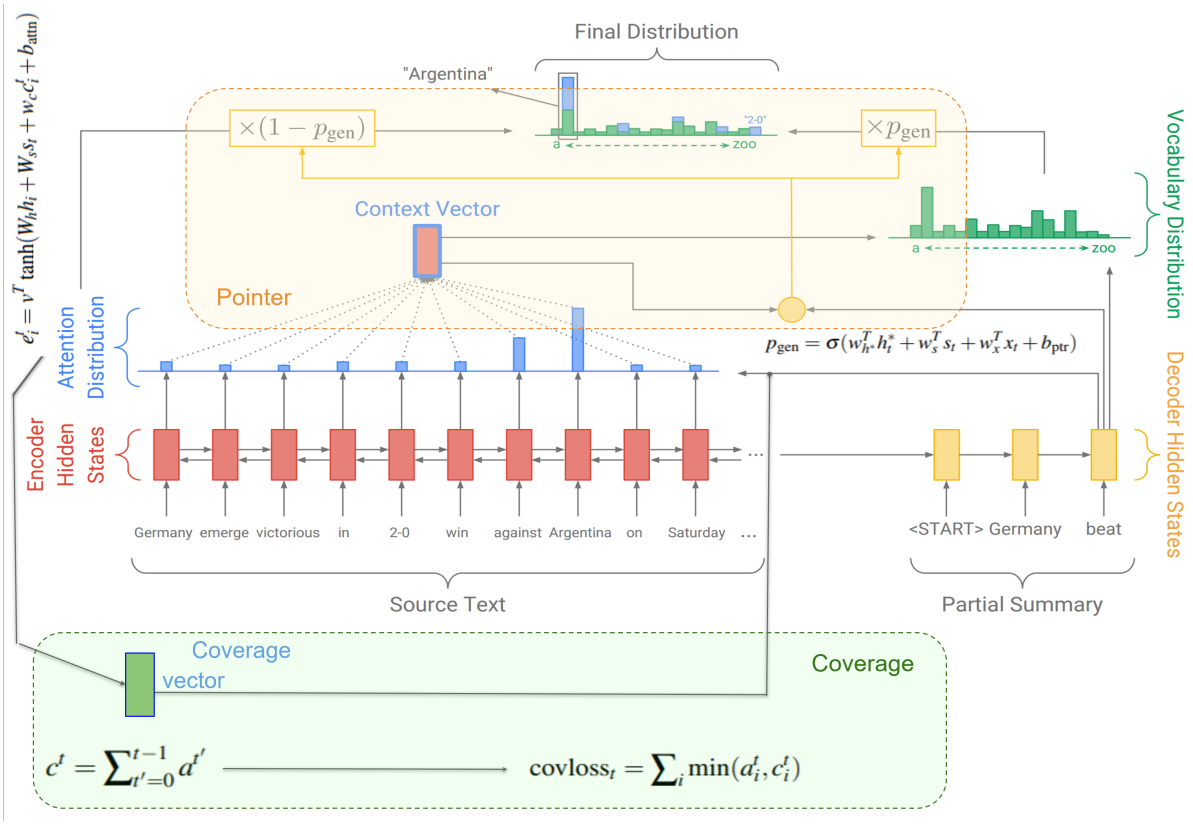

Figure 1: Overall structure of the model: Attentional seq2seq with pointer and coverage. The figure is modified from original figure in (See et al., 2017)

and coverage mechanism (2.3) (see full model in figure 1). The code for our models is available online. [1]

2.1 Attentional seq2seq

The baseline model is similar with seq2seq model in (Nallapati et al., 2016). **Encoder** is a single-layer bidirectional RNN which receives the tokens of the source (one-by-one) to produce a sequence of *encoder hidden states* h_i. The last hidden state (*source encoding*) is the first hidden state of the decoder. **Decoder** is a single-layer unidirectional RNN. On each step *t*, the decoder (with *state* s_t) receives the word embedding of the previous word and produce the vocabulary distribution. In short, decoder is a *conditional language model*, conditioned on the *source encoding*.

$$P_{vocab} = softmax(V'(Vs_t + b) + b') \quad (1)$$

Seq2seq suffers from the bottleneck problem: we need to store a large amount of information in the last encoder hidden state. Attention is added to solve this problem. Attention is a distribution capturing the dependency between the current decoder hidden states and all the encoder hidden states. In our model, we use the attention in (Bahdanau et al., 2015):

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn}) \quad (2)$$

$$a^t = softmax(e^t) \quad (3)$$

Information from all encoder hidden states will be interpolated with the weight decided by the attention distribution, producing the context vector h_t^*

$$h_t^* = \sum_i a_i^t h_i \quad (4)$$

The decoder state is concatenated with this context vector before being fed to the linear layer in (4). The modified probability distribution:

$$P_{vocab} = softmax(V'(V[s_t, h_t^*]) + b) + b') \quad (5)$$

The target word is sampling from this vocab distribution; therefore, probability of a word w is $P(w) = P_{vocab}(w)$ The loss is calculated as sum over timesteps of negative log likelihood of the target word, normalized by number of timesteps.

$$L_t = -logP(w_t^*) \quad (6)$$

[1] https://github.com/john-mai-2605/CS492-Auto-Summarization

$$L = \frac{1}{T}\sum_{t=0}^{T} L_t \quad (7)$$

2.2 Pointer-Generator Network

The pointer networks proposed in (Vinyals et al., 2015) is hybridized with the baseline to solve the OOV and wrong factual detail problem. It is notable that sometimes it is adequate to just copy the word from the source. We introduce a new parameter *"generation probability"* p_{gen} as a soft switch between pointer (copy from the source) and generator (generate novel words). p_{gen} is learned from the context vector h_t^*, the decoder state s_t and the input x_i.

$$p_{gen} = \sigma(w_h^T h_t^* + w_S^T s_t + w_x^T x_t + b_{btr}) \quad (8)$$

For each source (or batch of source), the vocabulary is extended to include all the word in the source, called the *extended vocabulary*. With the probability p_{gen}, the network generates a word from vocab by sampling from vocab distribution, otherwise copy the word from the source by sampling from attention distribution. The final distribution is calculated as

$$P(w) = p_{gen}P_{vocab}(w) + (1-p_{gen})\sum_{i:w_i=w} a_i^t \quad (9)$$

2.3 Coverage

The coverage mechanism proposed in (Tu et al., 2016) is adapted by introducing a vector c_t (*coverage vector*) to capture the previously summarized information as the sum of attention distributions over all previous decoder timesteps:

$$c_t = \sum_{t'=0}^{t-1} a^{t'} \quad (10)$$

It will be used as extra input for the attention in equation (2). Basically we tell the attention its history.

$$e_i^t = v^T tanh(W_h h_i + W_s s_i + w_c c_i^t + b_{attn}) \quad (11)$$

The repitition is further penalized by a *coverage loss*:

$$L_t^{cov} = \sum_i min(a_i^t, c_i^t) \quad (12)$$

If the attention tries to attend to a word attended a lot before, the coverage loss will be large (both a_i^t and c_i^t are large). The coverage loss is interpolated with the original loss in (6) to yield the new loss function:

$$L_t = -logP(w_t^*) + \lambda L_t^{cov} \quad (13)$$

3 Dataset

To make the dataset, we used the preprocessed data introduced in (Nikolov et al., 2018). For *title-gen* task (generating title using abstract) the data is driven from MEDLINE. [2] MEDLINE contains scientific articles in XML format of about 25 million paper in the biomedical domain.

These XML files[3] are processed to pair of abstractive and title for *title-gen* task, and then applied the preprocessing step from MOSES statistical machine translation pipleline [4]. The length limit is 150-370 tokens for abstract, and 6-25 tokens for title. Statistic of the dataset is in Table 1.

The *Overlap* $o(\mathbf{x}, \mathbf{y}) = \frac{|\{\mathbf{y}\}\cap\{\mathbf{x}\}|}{|\{\mathbf{y}\}|}$ is the fraction of unique output tokens $\mathbf{y}$ that overlap with an input token $\mathbf{x}$ (excluding punctuation and stop words). As can be seen in Table 1, the overlaps are large in the dataset, indicating frequent reuse of words. The Repeat $e(\mathbf{s}) = \frac{\sum_i o(\overline{\mathbf{s}_i}, s_i)}{|\mathbf{s}|}$ is the average overlap of each sentence s_i in a text with the remainder of the text (where s_i denotes the complement of sentence $\overline{\mathbf{s}_i}$). Repeat measures the redundancy of content within a text: a high value indicates frequent repetition of content.

title-gen	*abstract*	*title*
Token count	245 ± 54	15 ± 4
Sentence count	14 ± 4	1
Sent. token count	26 ± 14	-
Overlap	$73\% \pm 28\%$	
Repeat	$44\% \pm 11\%$	-
Size(tr/val/test)	5‘000‘000/6844/6935	

Table 1: Statistics (mean and standard deviation) of the scientific summarization dataset: title-gen. Token/sentence counts are computed with NLTK.

4 Experimental Setup

4.1 Model selection

We used GRU as the RNN cell, as it has higher performance and works faster than LSTM.

We do not pre-train the word embedding. Based on our experiment, the pre-train word embedding (Word2Vec, GloVE) has a slightly higher performance and works almost two times slower.

We test the model with three different learning rate: 0.01, 0.001 and 0.0001. While learning rate

[2] https://nlm.nih.gov/databases/download/pubmed_medline.html
[3] Use https://titipata.github.io/pubmed_parser.
[4] github.com/moses-smt/mosesdecoder

0.01 does not decreases the loss and learning rate 0.0001 reduces the loss very slow, 0.001 is the optimal rate.

We use 20k vocabulary for 7K validation dataset, and 40k vocabulary for 70K subset of full dataset.

Unless specified by the embedding, the word embedding is 128-dimensional. We use 256-dimensional hidden state, coverage loss interpolation weight $\lambda = 1$.

Gradient clipping (clipped by norm 10) and smoothing terms (1e-38) were used to solve NaN problem.

4.2 Experiment

Two different abstractive models were trained. A seq2seq model similar to the one described by (Nallapati et al., 2016), with the exception of using GRUs instead of LSTMs, and a pointer-generator model similar to the one described by (See et al., 2017). The seq2seq model was trained for 30 epochs and fine-tuned with the coverage loss for 5 epochs on a GeForce GTX Titan X GPU. The training data consisted of 70.000 randomly extracted samples from the full dataset. The pointer-generator model was trained for 40 epochs and fine-tuned for 10 epochs with the coverage loss on a Tesla K80 GPU in Google Colab. The validation dataset consisting of roughly 7000 samples was used for training (the test dataset was used for validation). For both models the Adam optimizer with learning rate 0.001 was used and 128-dimensional word embeddings were trained from scratch.

5 Results

5.1 Quantitative results

Our results are presented in table 2. To measure the quality of a summary the F_1-score of the ROUGE metric (Lin, 2004) is used. Here R-1 corresponds to the word-overlap, R-2 to the bi-gram overlap and R-L to the longest common sequence. As a baseline model the first sentence of the abstract is used as the title, denoted *lead-1*.

Table 2 shows that all of (Nikolov et al., 2018) models outperform the *lead-1* baseline substantially within all three ROUGE metrics. Internally, the *char2char* and *subword-conv* models perform very similarly, while the *seq2seq-lstm* model has a worse performance. One thing to note is that the *lead-1* baseline has a much higher token count with a large spread, indicating that it tends to generate too long summaries when compared to the abstractive summarization models.

Our models all outperform the *lead-1* baseline in the R-1 and R-L metrics, but all have very poor performance on the R-2 metric. This indicates that while the models are able to pick out key-phrases they do not produce semantically meaningful summaries. As mentioned in section 4 due to a lack of computational resources and time restrictions our models are trained on significantly less data than (Nikolov et al., 2018). This severe reduction of the dataset is most likely one of the main drivers for the bad performance within the R-2 metric. Somewhat surprisingly the pointer-generator models, which are trained on 10 times less data, actually out perform the seq2seq models trained on the larger dataset, indicating the strength of using a pointer to copy words directly from the source text.

5.2 Qualitative results

While quantitative metrics, such as ROUGE, are useful for evaluating overall performance of a model they often don't provide any insights into how the model works, and why it might or might not have a good performance. Thus, two qualitative results are presented in Example 1 and 2. Both examples are generated using the pointer-generator network with coverage, but in Example 1 the model is evaluated on unseen test data, while the evaluation is performed on previously seen training data in Example 2. Green indicates that the word appeared in the source text, while red indicates that the word is fabricated. Stop-words such as *in* and *the* have been omitted from this highlighting. The blue shading illustrates the generation probability, with dark blue indicating a high generation probability.

In example 1 it can be observed that quite a few erroneous words are generated, and while smaller phrases make sense the overall summary lacks coherence. However, a few key concepts regarding the study are captured, such as *lymph nodes* and *chemotherapy*. The correct words are mainly extracted using the pointer, while the generator seems to generate erroneous words most of the time. This indicates that the generator module most likely needs more training, while the decoder and pointer modules seem to have an acceptable performance.

In example 2 it can be seen that the model relies heavily on the pointer in order to extract correct information from the source text, while only using the generator module for words that bind sentences

Model	R-1	R-2	R-L	Token count
lead-1	0.316	0.136	0.310	28 ± 14
Nikolov et al. 2018				
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
Ours				
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

Table 2: First three models are results reported by (Nikolov et al., 2018). Four last models are replications of the models proposed by (See et al., 2017), and applied to the scientific articles dataset. 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 test dataset).

together. Another thing to note is that the generation probabilities are very binary. This in combination with the heavy use of the pointer could indicate overfitting, most likely due to the small size of the training data.

Example 1

91 acute and lymph node detection in adults with chemotherapy regimens in a admitted with a hepatic cervical brachytherapy

R-1=0.2286 **R-2**=0.0 **R-L**=0.2931

Target 1

radiation therapy and concurrent cisplatin in management of locoregionally advanced nasopharyngeal carcinomas

Example 2

trends in entropy production during ecosystem development in the amazon development

R-1=0.9231 **R-2**=0.8333 **R-L**=0.9355

Target 2

trends in entropy production during ecosystem development in the amazon basin

6 Discussion

Implementation of a pointer-generator network for summarization of scientific articles is not a straightforward task. Especially, the large corpus considered in this case means that not only knowledge within machine learning theory is required, but also knowledge on efficiently implementing such models in well-known frameworks such as `tensorflow` or `pytorch`. While the performance of our pointer-generator model did not surpass that of (Nikolov et al., 2018), we're confident that given more time and computational resources similar or better results could be obtained.

Another interesting thing to note is that with the need for interpretable models becoming ever more prevalent the pointer-generator network will most likely only become more relevant. The coverage vector provides a nice explanation for what parts of the input the model attended to, while the generation probability vector shows which parts of the network were used for generation each word in the summary. In conjunction these two vectors provide a nice explanation for the underlying decisions of the model.

The pointer network and coverage mechanism is very handy and applicable to many models.

The pointer only adds a "soft switch" between pointer/generator and can be applied to any generation algorithms. The coverage can be applied on top of both model (with or without pointer) and can be train as a separate training phase (train the model without coverage first and then add coverage to continue training). These two mechanisms may become helpful to combine with other models in the future.

The inclusion of pointer-generator network makes a delicate balance between extractive and abstractive method of summarization. The functionality of pointer is reasonable, because human also use some words from the source in stead of trying to produce a new word every time. The pointer becomes even more effective in case of scientific article, because there are a large amount of unfamiliar (OOV) words and contents related to domain knowledge. In these cases, pointer takes precedence over generator to copy content from source text and produce a good summary.

In (See et al., 2017), the author argues that lead-3 baseline is very strong for news article because most of the important information are in the first few sentences. Our model beats the lead-1 as least in R-1 and R-L metric, meaning that lead-1 is not a strong baseline for abs2title of scientific articles. It is reasonable since the main content of scientific article usually cannot encapsulated in the first sentence. Our result shows this nature of the dataset.

Finally, as mentioned by (See et al., 2017) the ROUGE metric does not always provide a fair assessment of summaries produced by abstractive models. The use of abstraction means that the model can generate novel words that do not appear in the source text, in order to provide shorter and more human-like summaries. However this quality is not captured by the ROUGE metric, which might instead punish the use of a novel word if it does not appear in the target summary. One way to remedy this issue would be to use multiple target summaries, preferable written by different people, and use the best ROUGE score among these summaries as the performance indicator.

References

Sumit Chopra Alexander M Rush and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Empirical Methods in Natural Language Processing*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate.

Oriol Vinyals Ilya Sutskever and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems*.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *International ACM SIGIR conference on Research and development in information retrieval*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. page 10.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond.

Nikola I. Nikolov, Michael Pfeiffer, and Richard H. R. Hahnloser. 2018. Data-driven summarization of scientific articles.

Chris D Paice. 1990. Constructing literature abstracts by computer: techniques and prospects. *Information Processing Management*, 26(1):171–186.

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 3–21. Springer.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks.

Michael Auli Sumit Chopra and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *North American Chapter of the Association for Computational Linguistics*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks.

Sanja Fidler Wenyuan Zeng, Wenjie Luo and Raquel Urtasun. 2016. Efficient summarization with read-again and copy mechanism.

A Appendices

A.1 Supplementary material

Source 1

radiation therapy in combination with chemotherapy in the management of locoregionally advanced nasopharyngeal carcinomas is evaluated in an attempt to improve locoregional response , reduce locoregional failure and reduce systemic failure . the current study was designed to investigate radiation therapy and concurrent cisplatin in this context . from 1992 through 1997 , 70 patients with locoregionally advanced nasopharyngeal carcinomas were treated with radiation therapy and concurrent cisplatin . external beam radiation dose was 60 gy for t1 , t2 and t3 tumors , 70 gy for t4 tumors and 70 gy for metastatic cervical nodes . an intracavitary brachytherapy boost (10 gy) was applied for t1 , t2 and t3 tumors . cisplatin (30 mg / m2) was administered weekly during external beam radiation therapy . locoregional complete response was achieved in 63 patients , locoregional failure was observed in 4 patients and systemic failure was observed in 15 . n-stage predicted systemic failure . overall survival , locoregional failure-free survival and systemic failure-free survival were 63 % , 79 % and 75 % , respectively , at three years . grade 3 acute skin toxicity was observed in 2 patients , grade 3 acute mucous membrane toxicity was observed in 6 and grade 3 acute hematological toxicity was observed in 2 patients . despite improved locoregional response , reduced locoregional failure and improved survival with radiation therapy and concurrent cisplatin , systemic failure remains prevalent for locoregionally advanced nasopharyngeal carcinomas .

Source 2

understanding successional trends in energy and matter exchange across the ecosystem-atmosphere boundary layer is an essential focus in ecological research ; however , a general theory describing the observed pattern remains elusive . this paper examines whether the principle of maximum entropy production could provide the solution . a general framework is developed for calculating entropy production using data from terrestrial eddy covariance and micrometeorological studies . we apply this framework to data from eight tropical forest and pasture flux sites in the amazon basin and show that forest sites had consistently higher entropy production rates than pasture sites (0.461 versus 0.422 w m (-2) k (-1) , respectively) . it is suggested that during development , changes in canopy structure minimize surface albedo , and development of deeper root systems optimizes access to soil water and thus potential transpiration , resulting in lower surface temperatures and increased entropy production . we discuss our results in the context of a theoretical model of entropy production versus ecosystem developmental stage . we conclude that , although further work is required , entropy production could potentially provide a much-needed theoretical basis for understanding the effects of deforestation and land-use change on the land-surface energy balance .