ISSN 2395-8618

A Neural Topic Summarizer: Using Topic Model for Enrichment of Abstractive Summaries

Giovanni Siragusa and Luigi Di Caro

***Abstract*—Nowadays, Neural Networks are widely used for *abstractive* and *extractive* summarization, since they are able to create human-like summaries. To the best of our knowledge, few existing neural models for summarization use a priori word features such as POS tags, which require time to be generated. In this paper, we present a model called *NeTSumm* that merges the Sequence-to-Sequence model with topics, computed *on-the-fly*, to extract hidden thematic structures which influence the generation of the summaries. Despite our model did not reach state-of-the-art results, it was able to better discover actual relations between words.**

***Index Terms*—summarization, neural networks, topic modelling, natural language processing.**

I. Introduction

Text summarization is the task of compressing a document in a shorter form, while preserving all relevant information. There exist two approaches: *extractive* methods, where sentences are extracted from the input document and rearranged to create the summary, and *abstractive* ones that may use words outside of the document vocabulary. While abstractive summarization is more challenging and scientifically interesting, extractive approaches received large attention to date [1], [2], as they only require some score function to be applied to the sentences in the document, later selecting those with the highest scores for the summary generation.

Recently, with the explosion of neural networks models for Natural Language Processing, neural-based abstractive summarization increasingly emerged. In particular, all proposed models fall under the encoder-decoder framework: the encoder transforms the input text in an abstract representation, which is then used by the decoder to generate the output summary, in a word-by-word fashion, using a probability distribution over the vocabulary. Rush et al. [3] used an attention-based encoder followed by a neural language model [4] to read the input text and generate the summary. Chopra et al. [5] substituted the language model with a RNN-based one. Finally, Nallapati et al. [6] adopted the Sequence-to-Sequence model [7] for the summarization task. Those models also use the attention approach proposed by Bahdanau et al. [8] to select relevant portions of the input text, using them as context to generate the "next word" in the summary. In details, the encoder process each word in input and generate a vector representation; then, the decoder, at each step, attends to those vectors to select those ones that are relevant for the generation of the output word.

However, summarization models suffer from two drawbacks: (i) networks may generate summaries containing repetitions of identical words or sentences (this problem has been called *odd-generation* [9]); (ii) they are unable to deal with out-of-vocabulary (OOV) words and rare words [10]. The adoption of Neural Machine Translation coverage methods [11,12,13,14] and pointer networks [10,15] has mitigated such issues.

To the best of our knowledge, few existing models for summarization use global features to improve the knowledge of the network about the input text. For instance, Nallapati et al. [6] used word features such as TF (Term Frequency), IDF (Inverse Document Frequency), POS (Part-Of-Speech) tags and NEs (Named Entities) as enrichment. The drawback of this approach is the time-cost of computing the features.

In this paper, we believe that a faster approach to enrich the knowledge of the input text is to use global semantic features that are extracted *on-the-fly* by the neural network. In detail, we here propose the use of topics extracted by the neural topic model proposed in [11]. Our idea arises from the fact that documents are composed of several latent thematic structures (see Figure 1) that carry the semantic information of different parts of the text. Such extracted features could be used both in the decoder and the attention layer to improve the quality of the final summaries. We called this model *NeTSumm* (Neural Topic Summarizer).

Harry Potter villain Ralph Fiennes has been spotted paying a late night visit to a Thai massage parlour - but appeared to have a little trouble settling the bill. The actor, 55, who plays the reviled Lord Voldemort in the cult wizardry films, spent about an hour in the Soho venue in London after slipping in at 10.45pm. But his visit to the Boonsawad Thai Spa on Thursday did not go off without a hitch.

Fig. 1. The text is a small excerpt taken from cnn website https://www.cnn.com. The colors represent different possible topics unraveled from the text. Those topics could be used to guide the summarization model

The remain of this paper is structured as follows: in Section II we presents the state-of-the-art for Neural Abstractive Summarization, comparing our system with existing ones; in Section III we present the baseline model; Section IV describes the topic model and its integration to the Sequence-to-Sequence model. Section V reports the results that we

Manuscript received on 07/01/2018, accepted for publication on 19/03/2018.

The authors are with the Computer Science Department, University of Turin, Italy ({siragusa, dicaro}@di.unito.it).

ISSN 2395-8618

obtained on CNN/Dailymail dataset [12]. Finally, the article concludes in Section VI.

II. Related Works

To the best of our knowledge, the first model that used Topic Modelling [13] in summarization task is Topiary [14], which combined the topics with the generated summary. Topiary has been used as comparison baseline for many neural summarization model, such as [3,6,20]. Our model resembles Topiary since both use topics to generate summaries, with the difference that we directly use the topic distribution to guide the network. Our use of topics could be seen as a way to enrich encoder information, where topics represent global semantic information of the input text.

This is not the first research work that enriched encoder information to improve the quality of the generated summaries. For instance, Nallapati et al. [6] explored different features as Part-Of-Speech tags, Name Entity tags, Term Frequency and Inverse Document Frequency. The major drawback of Nallapati et. al.'s work is the time required to obtain those features for large corpora, like those ones used for neural summarization.

Other models, instead of using further features, focused to use reinforcement learning [15] or to combine the decoder probability distribution with the input distribution generated by the pointer network [16]. See et al. [10] assumed that the attention approach could be seen as a pointer network and proposed a mixture model that combines attention distribution with the vocabulary distribution. Paulus et al. [15] tried to contrast *exposure bias* [17] using reinforcement learning.

Both models use coverage to remove duplicated words and sentences. Coverage, proposed by Kohen [18] for machine translation, has been adapted for Sequence-to-Sequence models by Tu et al. [19] and by Mi et al. [20]. Those works defined coverage as a vector that stores the previous attention distribution (i.e., it is an attention history) to prevent repeated focusing on the same input locations. In details, the coverage value of a word is used as further feature to calculate word score (see Equation 1). Recently works explored different uses of coverage to distract the decoder [21] or to modify attention weights [22].

III. Model

Let $\mathbf{x} = [x_1, x_2, \ldots, x_M]$ be a sequence of M input tokens, and $\mathbf{w} = [w_1, w_2, \ldots, w_N]$ be a sequence of N target (summary) tokens, with $N < M$. Let all tokens belong to a vocabulary V, with $|V| = K$. Our model, similar to the one proposed by [6] and depicted in Figure 2, consists of a Bidirectional LSTM Encoder and an attention-based LSTM Decoder.

At each step i, the encoder is fed with the input tokens x_i and produces an encoded state h_i. Similar to the encoder, on each step t the decoder (a single unidirectional LSTM) receives in input the embedding of the previous word[1], the decoder state s_t and the context vector c_t, using them to emit a word in output (the network emits a probability distribution over the vocabulary that is used to select the next word).

The context vector is calculated as in [10]:

$$e_j^t = v^T \, tanh(W_e \, [s_t || h_j || cov_j^t] + b_e),$$

$$a_j^t = \frac{\exp(e_j^t)}{\sum_{k=1}^{M} \exp(e_k^t)}, \tag{1}$$

$$c_t = \sum_{k=1}^{M} a_k^t \, h_k,$$

where $||$ is the concatenation operator.

The attention approach could be seen as a probability distribution over the sequence of encoded states, that gives to the network the relevant portions of the input text to use in the emission of the current word. In Equation 1, v, W_e and b_e are learnable parameters of the model, v^T is the transpose of vector v, and cov_j^t contains the sum of the previous attention scores to contrast the odd-generation problem [9]:

$$cov_j^t = \sum_{t'=1}^{t-1} a_j^{t'}. \tag{2}$$

In our model, we adopted the mixture model proposed by [10], which combines the probability of a vocabulary word with its attention score, to deal with the Out-Of-Vocabulary problem and the rare words. Thus, the probability of emitting a word y_i at step t is computed as follow:

$$\hat{h}_t = tanh(W_c \, [y_{t-1} || c_t || s_t] + b_c),$$
$$P(y_i^t) = \beta \, softmax(W_o \, \hat{h}_t + b_o) + (1 - \beta) \sum_{w=y_t} a_w^t, \tag{3}$$

where W_o, W_c, b_o and b_c are learnable parameters and y_{t-1} is the word embedding of the previous emitted word. β is a value within the range $[0, 1]$ used to mix the two distributions and it is calculated through Equation 4, where W_b and b_b are leanable parameters and σ is the sigmoid function:

$$\beta = \sigma(W_b \, [y_{t-1} || c_t || s_t] + b_b). \tag{4}$$

We trained our model to minimize the sum of the negative log-likelihood of the sequence of target words w^*:

$$loss = \frac{1}{N} \sum_{t=1}^{N} -log \, P(w_t^*). \tag{5}$$

IV. Topic Model

As described in the introduction, our idea is to extract topics from the input document, which could be seen as global semantic features that may lead the decoder to explore further relevant parts of the document to generate the output summary.

For our model, we used the neural topic model proposed in [11], with two main differences:

— we used a document representation generated by the neural network instead of the embedding of the input document words. In this way, we leave to the network the task to filter irrelevant information from the input document;

[1] During training it is the embedding of w_{i-1}, while in testing it is the embedding of the previous word emitted by the decoder.

ISSN 2395-8618

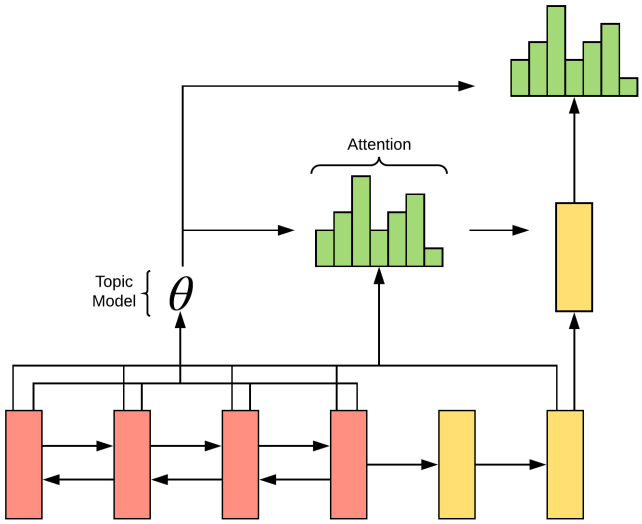

Fig. 2. The proposed neural network model. The first four (red) rectangles represent the encoder states, while the last two (yellow) rectangles represent the decoder states

— we used the generated topics as features in the tanh-layer of Equation 3, allowing the network to use the global semantic context to generate the next word.

Given the sequence to encoded states h, we computed the document representation as follows:

$$D = tanh(W_{doc} \frac{1}{M} \sum_{j=1}^{M} maxout(W_h\ h_j)). \tag{6}$$

The underlying idea in Equation 6 is to select those features that are relevant in the construction of the document representation. First, the weight matrix W_h is used to unveil the features in the i-th encoded state h_i. Relevant features are extracted by the *maxout* function. Finally, the document representation D is computed projecting the average of the vectors. Once D is calculated, it is passed as input to the topic model to generate the topic vector θ:

$$\begin{aligned} \mu &= W_\mu\ D, \\ \log(\sigma^2) &= W_\sigma\ D, \\ \theta &= \mu + \sigma * \epsilon, \end{aligned} \tag{7}$$

where $\epsilon \sim N(0, I)$ is the reparameterization trick, W_μ and W_σ are two learnable parameters.

The set of topics θ is used to affect e_j^t and $p(y_i)$. In details, we rewrite Equation 1 and the tanh-layer of Equation 3 as follows:

$$\begin{aligned} e_j^t &= v^T\ tanh(W_e\ [s_t||h_j||cov_j^t||\theta] + b_e), \\ \hat{h}_t &= tanh(W_c\ [y_{t-1}||c_t||s_t||\theta] + b_c). \end{aligned} \tag{8}$$

Since the topic model requires to optimize the marginal likelihood that is intractable [11], we used variational inference. We transformed Equation 3 into a variational objective function, also called estimated lower bound (ELBO), as follows:

$$loss = \frac{1}{N} \sum_{t=1}^{N} -log\ P(w_t^*) - KL(q(\theta)||p(\theta)), \tag{9}$$

where KL is the Kullback-Leibler divergence between the variational distribution $q(\theta)$ and the topic probability distribution $p(\theta)$.

V. Experiments

We trained and tested our model on *CNN/DailyMail* dataset [12], which contains online articles paired with multi-sentence summaries. We used the script supplied by [6] to obtain the same version of their dataset that consists in 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs. Each article has 781 tokens on average, while each summary has 3.75 sentences and 56 tokens on average. Differently from [6], we used an *non-anonymized* version of the dataset[2].

Our models have 256 dimensional hidden layers and 128 dimensional embedding layer. Following [10], we used a small vocabulary comprising of 50k tokens for both source and target. For the topic model, we tested with 50 and 100 topics (θ vector dimension). The models are trained using AdaGrad [24] with a learning rate of 0.15. We also used gradient clipping with a gradient norm of 2.0 and dropout [25] with probability 0.2 (keep probability of 0.8) to improve model generalization. We used the loss on the validation set for early stopping.

During training and testing, we truncated the article length to 400 tokens and summary length to 100 tokens in order to speed up both the convergence of the model and the generation of the summaries. In details, we started the training with an article length of 200 tokens and a summary length of 50 tokens, and we progressively increased such values. The

[2]Please, see [23] for problem regarding the anonymized version.

ISSN 2395-8618

training was performed on a single GPU K40m, with a batch size of 16. At testing time, we used a beam size of 4.

We trained our baseline model for about 470,000 iterations (28 epochs), and our topic summarization model for about 370,000 iterations (22 epochs). The training of the baseline model took about 4 days of computation, while the topic one took 3 days of computation.

A. Results

In this section we will report Rouge scores (Rouge-1, Rouge-2 and Rouge-L) [26] of our models. We compare our models with See et al. [10]'s pointer generation model and Nallapati et al. [6]'s abstractive model.

Table I reports the results of our models, computed using pyrouge package[3]. From the table we can see that the models with topics fall behind the baseline, which is close to the Pointer-generation one. We can also see that the model with 100 topics surpassed the one with 50 topics, meaning that it was able to capture more hidden thematic structures of the text. We think that the loss of performance is due to the Kullback-Leibler divergence that forced the model to principally approximate the two probability distributions instead of learning how to generate the summaries. Furthermore, the models with topics suffered of gradient explosion: after 350,000 training steps, the models started to return an infinite loss both in training and evaluation.

TABLE I
THE TABLE REPORTS ROUGE-1 (R-1), ROUGE-2 (R-2) AND ROUGE-L (R-L) SCORES OF THE MODELS. THE SYMBOL ‡ MEANS THAT THE MODEL WAS TRAINED AND EVALUATED ON THE ANONYMIZED DATASET, AND WE CANNOT STRICTLY COMPARE IT WITH OUR MODELS

Model	R-1	R-2	R-L
Abstractive Model‡	35.46	13.30	32.65
Pointer-generation	36.44	**15.66**	**33.42**
NeTSumm without topics	**36.73**	**15.66**	26.31
NeTSumm with 50 topics	33.88	12.59	24.20
NeTSumm with 100 topics	34.18	13.00	24.52

We report some generated summaries of *NetSum without topics* and *NetSum with topics*. Table II reports the text of a source document, the human generated summary and the summaries generated by our three models. As we can see, *NeTSumm with 100 topics* is the one that used the words "*CDC*" and "*samples*" that are not present in the other summaries. We can also see that the automatic summaries contain repetition of words (like "*listeria contamination*" in *NeTSumm with 50 topics*). Other examples of summaries are reported in Table III and Table IV.

Finally, we explored how topics influenced the attention scores. We saw that topics increase the energy (e^t values in Equation 1) of previous attended words, i.e. they force the network to focus on words that are not relevant at the current timestep. This caused repetition of previously generated n-grams, and, in certain cases, wrong summaries. We reported an example in Figures 3 and 4.

Figure 3 shows the attention scores (for a portion of the summary) of the model without topics, while Figure 4 shows the attention scores when 100 topics are added. Looking at the two figures, we can see that the topics forced the model to copy again "tree", generating "rope from a tree near a tree" instead of "rope from a tree near a student union".

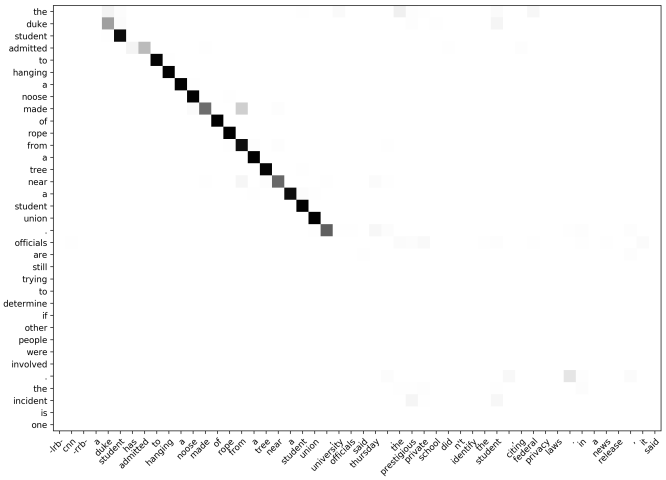

Fig. 3. The image shows the attention distribution for some timesteps. Dark colors represent high attention scores

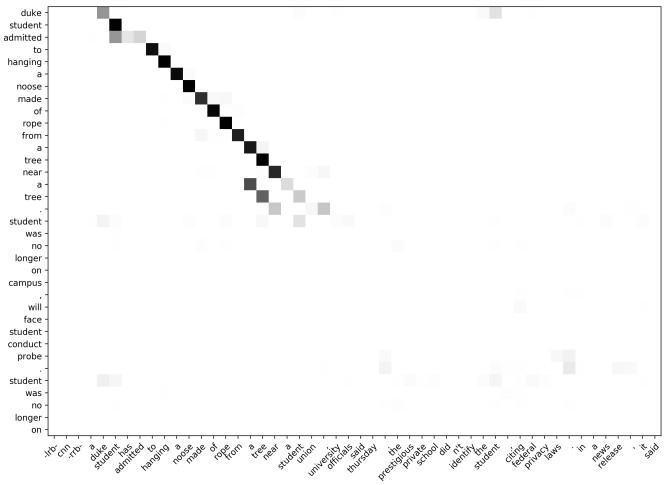

Fig. 4. The image shows the attention distribution, calculated adding topics, for some timesteps. Dark colors represent high attention scores

VI. CONCLUSION

In this paper, we presented a Sequence-to-Sequence model that generates topics from the text and employs them in the generation of the summaries. In Section V, we discovered that the use of topics does not improve the final summaries in terms of Rouge scores. Despite the non state-of-the-art results, our model was able to better unravel semantic relations between words, and using them in the generation of the summaries. Those information cannot be captured by the Rouge metric, which is based on the overlap of n-grams.

[3]pypi.python.org/pypi/pyrouge/0.1.3

ISSN 2395-8618

TABLE II
THE TABLE SHOWS THE SUMMARIES GENERATED BY OUR SYSTEMS, AND THE ORIGINAL ONE CREATED BY HUMAN

Source Document
blue bell ice cream has temporarily shut down one of its manufacturing plants over the discovery of listeria contamination in a serving of ice cream originating from that plant . public health officials warned consumers friday not to eat any blue bell-branded products made at the company 's broken arrow , oklahoma , plant . that includes 3-ounce servings of blue bell ice cream from this plant that went to institutions in containers marked with the letters o , p , q , r , s or t behind the coding date . [...]
Human Summary
a test in kansas finds listeria in a blue bell ice cream cup . the company announces it is temporarily shutting a plant to check for the source . three people in kansas have died from a listeria outbreak .
NeTSumm without topics
public health officials warned consumers friday not to eat blue bell-branded products . that includes 3-ounce servings of blue bell ice cream from this plant . the company is shutting down the broken arrow facility " out of an abundance of caution " to search for a possible cause of contamination .
NeTSumm with 50 topics
listeria contamination discovery of listeria contamination in a serving of ice cream originating from that plant . public health officials warn consumers not to eat any blue bell-branded products . the company is shutting down the broken arrow facility out of abundance of caution .
NeTSumm with 100 topics
cdc : listeria samples taken from patients infected in kansas outbreak . company is shutting down broken arrow facility " out of an abundance of caution " the company is shutting down the broken arrow facility .

TABLE III
ANOTHER EXAMPLE OF GENERATED SUMAMRIES. IN *NeTSumm with 100 topics* WE CAN SEE THAT THE MODEL REPEATED A SENTENCE

Source Document
marseille , france -lrb- cnn -rrb- the french prosecutor leading an investigation into the crash of germanwings flight 9525 insisted wednesday that he was not aware of any video footage from on board the plane . marseille prosecutor brice robin told cnn that " so far no videos were used in the crash investigation . " he added , " a person who has such a video needs to immediately give it to the investigators . " robin 's comments follow claims by two magazines , german daily bild and french paris match , of a cell phone video showing the harrowing final seconds from on board germanwings flight 9525 as it crashed into the french alps . all 150 on board were killed . [...]
Human Summary
marseille prosecutor says " so far no videos were used in the crash investigation " despite media reports . journalists at bild and paris match are " very confident " the video clip is real , an editor says . andreas lubitz had informed his lufthansa training school of an episode of severe depression , airline says .
NeTSumm without topics
french prosecutor says he was not aware of video footage from on board plane . robin 's comments follow claims by magazines , german daily bild and french paris match . video shows harrowing final seconds from on board germanwings flight 9525 .
NeTSumm with 50 topics
" person who has such a video needs to immediately give it to the investigators , " he says . robin 's comments follow claims by two magazines , german daily bild and french paris match . cell phones have been collected at the site , french prosecutor says .
NeTSumm with 100 topics
robin 's comments follow claims by two magazines , german daily bild . robin 's comments follow claims by two magazines , german daily bild . robin 's comments follow claims by two magazines , german daily bild .

TABLE IV
ANOTHER EXAMPLE OF GENERATED SUMMARIES IN WHICH IT IS PRESENT A REPETITION OF SENTENCES. FROM THE SUMMARIES, WE CAN SEE THAT *NeTSumm with 50 topics* PRODUCED A COMPLETELY UNUSABLE SUMMARY, WHILE *NeTSumm with 100 topics* WAS ABLE TO USE THE WORD *red cross*

Source Document
-lrb- cnn -rrb- seventy years ago , anne frank died of typhus in a nazi concentration camp at the age of 15 . just two weeks after her supposed death on march 31 , 1945 , the bergen-belsen concentration camp where she had been imprisoned was liberated – timing that showed how close the jewish diarist had been to surviving the holocaust . but new research released by the anne frank house shows that anne and her older sister , margot frank , died at least a month earlier than previously thought . researchers re-examined archives of the red cross , the international training service and the bergen-belsen memorial , along with testimonies of survivors . [...]
Human Summary
museum : anne frank died earlier than previously believed . researchers re-examined archives and testimonies of survivors . anne and older sister margot frank are believed to have died in february 1945 .
NeTSumm without topics
anne frank died of typhus in nazi concentration camp at age of 15 . anne and her older sister died at least a month earlier than previously thought . anne and her older sister died at least a month earlier than previously thought .
NeTSumm with 50 topics
anne frank , her sister , margot frank , died in nazi concentration camp in 15 weeks ago . anne frank , her sister , margot frank , died at least a month earlier , researchers say . in 1944 , anne and margot probably did not survive to march 1945 .
NeTSumm with 100 topics
researchers find archives of the red cross in 1944 . anne and margot frank died at least a month earlier than previously thought . typhus is an infectious disease caused by lice that breaks out in places with poor hygiene .

As a future work, we want to add Warm-Up, a method that multiplies the Kullback-Leibler divergence by a parameter λ that is gradually increased from 0 to 1. In this way, the model will be able first to learn how to generate a summary given the input text, and then how to extract the topics to improve the summaries. We will also remove the topics from the attention approach.

ISSN 2395-8618

REFERENCES

[1] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017.

[2] H. Saggion and T. Poibeau, "Automatic text summarization: Past, present and future," in *Multi-source, multilingual information extraction and summarization*. Springer, 2013, pp. 3–21.

[3] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 379–389.

[4] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[5] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 93–98.

[6] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016, pp. 280–290.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] S. Kiyono, S. Takase, J. Suzuki, N. Okazaki, K. Inui, and M. Nagata, "Source-side prediction for neural headline generation," *arXiv preprint arXiv:1712.08302*, 2017.

[10] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2017, pp. 1073–1083.

[11] A. B. Dieng, C. Wang, J. Gao, and J. Paisley, "Topicrnn: A recurrent neural network with long-range semantic dependency," in *International Conference on Learning Representations*, 2017.

[12] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[14] D. Zajic, B. Dorr, and R. Schwartz, "Bbn/umd at duc-2004: Topiary," in *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, 2004, pp. 112–119.

[15] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017.

[16] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[17] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *International Conference on Learning Representations*, 2016.

[18] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[19] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 76–85.

[20] H. Mi, B. Sankaran, Z. Wang, and A. Ittycheriah, "Coverage embedding models for neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 955–960.

[21] Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang, "Distraction-based neural networks for document summarization," *arXiv preprint arXiv:1610.08462*, 2016.

[22] B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah, "Temporal attention model for neural machine translation," *arXiv preprint arXiv:1608.02927*, 2016.

[23] D. Chen, J. Bolton, and C. D. Manning, "A thorough examination of the CNN/Daily Mail reading comprehension task," in *Association for Computational Linguistics (ACL)*, 2016.

[24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.