ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Modular Length Control for Sentence Generation

Katya Kudashkina[1,2] and Peter Wittek[2,3] and Jamie Kiros[4] and Graham W. Taylor[1,2]

1- University of Guelph - Engineering Dept.
50 Stone Rd E, Guelph, ON N1G 2W1 - Canada

2- Vector Institute for Artificial Intelligence
661 University Ave., Suite 710, Toronto, ON, M5G 1M1 - Canada

3- University of Toronto
105 St George St., Toronto, ON, M5S 3E6 - Canada

4- Google Brain
111 Richmond St. W. #12, Toronto, ON, M5H 2G4 - Canada

Abstract. Generating summary-sentences with preserved meaning is important for the summarization of longer documents. Length control of summary-sentences is challenging as sentences cannot simply be cut at the desired length; they must be complete and preserve input meaning. We propose a *modular framework* for length control of generated sentences: based on sequence-to-sequence models, powered by a two-stage training process involving a *summarizer* that is trained without explicit length control and a *stylizer* that is fine-tuned on the output of the summarizer. Our solution achieves the performance of existing models for controlling generated sentence length but light in implementation and model complexity.

Automatically generated, accurate summaries are becoming critically important due to the sheer amount of information available on any given subject. Text summarization is a process to achieve this end by rewriting a sentence or a paragraph into a shorter one while retaining the meaning. Summarization is classified as either *extractive*, where the output sentence is comprised directly from fragments of the input, or as *abstractive*, where the generated output contains text that may not necessarily be present in the input. Abstractive summarization is more natural and closer to what a human would do: humans can paraphrase a story or an article and express it in their own words. A number of applications of summarization today impose a length limit on the desired output. For example, a reader may have very limited time and would prefer a shorter output. Or alternatively, the summary may be broadcast through a service with a hard character limit (e.g. Twitter, SMS). Summarization tasks can vary from the summary of a single document to the summary of multiple documents, with summaries varying in length. Our focus is on abstractive summarization at the sentence level.

Neural sequence-to-sequence models have proven to be successful in abstractive summarization. Until recently, these methods have not offered any means of explicit length control; allowing the model to implicitly define the length through the decoding process (e.g. beam search). Recent works have shown that explicit length control is possible through different architectural modifications. In this paper, we show that instead of complexifying the summarizer with length control constraints, it is possible to leave it unconstrained, and train a second module, which we call the stylizer, to impose length control.

We argue that modularity is important for two reasons. First reason is that decoupling the process of summarization from that of matching a target length allows for "specialist" rather than monolithic components. Our stylizer performs a well-defined, specific task, and can act on the output of *any* sentence generation model. The second reason is the potential for more variability of the control parameters. The stylizer could be replaced with another model that allows the control of different properties such as style and sentiment. Though it is beyond the scope of this paper, our design is ultimately motivated by the possibility of controlling several properties at once.

1 Related Work

Similar to other applications of deep learning to NLP, the solutions for summarization were gradually developed using increasingly complex architectures while inheriting some similar components. Many models for summarization were inspired by work in machine translation [1, 2, 3].

Furthermore, there has been research showing the efficient use of generative models for abstractive summarization [4]. Our goal in this work is to find a path through the use of existing neural models, while creating additional capabilities and yielding comparable performance, without introducing more architectural complexity.

Our model also builds upon prior research on *controllable* sentence generation. Until the seminal work of Kikuchi et al. [5], encoder-decoder models for summarization did not explicitly control the length of output sentences. Similar to that work, we incorporate the controllable property (here, length) within the decoder during training and testing. As a follow-up, Fan et al. [6] proposed a different way of incorporating length constraints by introducing additional inputs in the form of control variables. They also proposed a way to control other properties of interest: the inclusion of specific entities and source style.

2 Controlling Length

In this section, we describe our proposed architecture for abstractive summarization. At a high-level, first we train a summarizer without any length control on the training dataset $\mathcal{D}_1$. Next, we run the summarizer in evaluation mode on $\mathcal{D}_1$ to produce summaries of varying length, which are filtered by ROUGE score to form a secondary, synthetic dataset $\mathcal{D}_2$. Then a stylizer is trained on $\mathcal{D}_2$ to post-process the output of the summarizer. The stylizer uses a length control mechanism while maintaining high-quality summaries. The stylizer is independent from summarizer and does not need to access the summarizer's parameters: it only requires access to $\mathcal{D}_1$ and $\mathcal{D}_2$. We now describe each module in detail.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

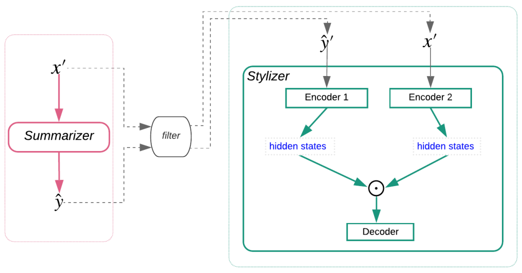

Fig. 1: Interaction of the summarizer and the stylizer: $\odot$ denotes the Hadamard product (element-wise multiplication) between the two hidden state vectors encoded by the stylizer.

2.1 Summarizer

We represent an input source sentence as a sequence of words $x = (x_1, x_2, ...x_{N_x})$, the sequence generated by the summarizer as $\hat{y} = (\hat{y}_1, \hat{y}_2, ...\hat{y}_{N_{\hat{y}}})$, and the ground truth sentence as $y = (y_1, y_2, ...y_{N_y})$. Our base model is a sequence-to-sequence model [7] with attention. It receives input sentence x, produces summary-sentence $\hat{y}$ and is trained to minimize the negative log-likelihood of the conditional probability $p(y|x)$ using the ground-truth summaries y. The conditional probability of the sentence is defined as a product of the probabilities of the next word in the sequence given all the previous words: $p(y|x) = \prod_{t=0}^{N} p(y_t|y_{<t}, x)$.

We use a bi-directional RNN (Bi-RNN) for both the encoder and decoder with the attention mechanism by Bahdanau et al. [2]. We train the summarizer as a standard encoder-decoder on the training dataset without any control parameters. As shown in Figure 1, the summarizer could be replaced by *any* other sentence generation model.

2.2 Stylizer

The stylizer is also a Bi-RNN for both the encoder and decoder with the same attention mechanism. The stylizer treats the summarizer as a black box.

Further, we incorporate a controllable parameter into the stylizer: sentence length. Inspired by the *LenEmb* model by Kikuchi et al. [5], at each step t we encode the remaining length of the target sentence. However, in contrast to their work, we concatenate the remaining length not only with the input token, but also with the context vector c that is derived from the attention mechanism. Another difference with Kikuchi et al. [5] is that we control the remaining length of the sentence by the number of desired words, instead of bytes. We found it more intuitive for a user to set word length, but this is not a critical design choice with respect to performance.

To control the quality of the training data for the stylizer we use filters. Since the stylizer is trained on synthetic data produced by the summarizer, the filters ensure that it sees a "clean" subset. An intuitive view of this is of a teacher, pointing out examples where a student was mostly correct, and asking them to try again and improve. To produce $\mathcal{D}_2$ using the *summarizer*, we assess an output sentence $\hat{y}$ against the ground truth y by ROUGE-1 scores and filter out sentences with poor score to form examples $\hat{y\prime} = (\hat{y\prime}_1, \hat{y\prime}_2, \hat{y\prime}_3, ..., \hat{y\prime}_{N_{\hat{y\prime}}})$. We considered two kinds of filters described below.

Filter 1: We select sentences with scores exceeding 50, 73, 89 for the length of $\hat{y}$ in the ranges (20,30), (40,50), and (60,70) respectively. The ROUGE-1 score rapidly increases with sentence length, hence the need to increase the cut-offs.

Filter 2: We use a beam size of 3 and select the sentence with the maximum ROUGE score.

The secondary dataset $\mathcal{D}_2$ comprises pairs of input sentences: a long sentence x from the training corpus, and a filtered summary sentence $\hat{y\prime}$ produced by the summarizer, as well as the corresponding ground truth short sentence y from $\mathcal{D}_1$.

The stylizer encoder separately encodes both the long sentence x, and the source sentence $\hat{y\prime}$ from a short-sentences pair from $\mathcal{D}_2$. The corresponding hidden states are merged by taking the Hadamard product between the two. The final hidden vector is passed to the stylizer decoder.

Following the ideas of Juraska et al. [8], to enhance the quality of the final output, we select the top 5 best performing stylizers according to validation ROUGE score and ensemble them into a final model.

3 Experiments

Dataset. We used the Annotated English Gigaword corpus [9]. The dataset contains about 9.5M news articles sourced from various news services over the last two decades. The headline is paired with the first sentence of the source document to create an input summary pair of sentences. Following the protocol of Kikuchi et al. [5] the primary training dataset $\mathcal{D}_1$ consists of about 4M pairs after pre-processing. Our secondary dataset $\mathcal{D}_1$ contains 6K and 11.4K samples for filter 1 and 2 respectively.

Evaluation. Similarly to Kikuchi et al. [5], we used the standard evaluation set of DUC-2004 Task-1. The set consists of 500 source documents with their respective four human-written reference summaries. We used automatic evaluation and report F1-ROUGE scores for ROUGE-1, ROUGE-2, and ROUGE-L [10]. The ROUGE metrics compare an automatically produced summary against a human-produced set of references.

Implementation. We implemented the models in PyTorch 0.4.0. Training was performed with shuffled mini-batches of size 64, with vocabulary size of 20k. We fine-tuned hyperparameters via the Tree of Parzen Estimators algorithm using the HyperOpt library[1] with default parameters. We used Adagrad [11] for training the summarizer and AdaDelta [12] for training the stylizer. We

[1]https://github.com/hyperopt/hyperopt

	30 byte			50 byte			75 byte		
Model	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LenInit [5]	**14.31**	**3.27**	**13.19**	20.87	**6.16**	**19.00**	25.87	8.27	23.24
LenEmb [5]	14.23	3.21	13.02	20.78	5.97	18.57	26.73	**8.39**	**23.88**
Stylizer: no filter	13.56	2.82	12.49	20.88	4.85	18.33	26.68	6.17	22.81
Stylizer: filter 1	13.37	2.88	12.36	**21.01**	4.97	18.49	**27.67**	6.59	23.68
Stylizer filter 2	13.61	2.82	12.51	20.97	4.88	18.37	26.80	6.27	22.95
Stylizer (ensemble)	14.74	3.21	13.52	23.17	5.50	19.88	**30.34**	7.32	25.29
Fan et al. [6]*	21.81	7.51	21.05	25.39	8.38	23.37	30.00	10.27	26.43

Table 1: ROUGE scores for fixed length control models on DUC-2004 Task-1. Bold emphasize the best results, excluding ensemble and Fan et al. [6].
(*) We show the results of Fan et al. [6], however, it is not comparable to our model. To compare, our base model requires to be replaced by their foundational model.

initialized the hidden vectors of the encoders to zeros. The hidden vector of the decoders were initialized to the hidden output vectors of the encoders. When training the summarizer, we set the dropout rate to 0.09 and enforced teacher-forcing of 100%. Our sentence embedding size was 4,000, the embeddings of the decoder was 100, and the hidden vector size of RNNs was 700. Training was performed on a single Nvidia Titan XP GPU. We trained with gradient clipping 0.03 and 0.2 [13] and a learning rate of 0.00475 and 0.001 for the summarizer and the stylizer, respectively. We used the ROUGE 1.5.5 Perl script with the standard settings for ROUGE evaluation [10]. The summarizer was trained for 4 days. The stylizer was trained for about 1-2 hours, after being pre-trained on English Gigaword. We used a beam size of 3 in the beam search during final decoding.

Results. Our main results are presented in Table 1. The model by Kikuchi et al. [5] serves as the baseline. Our model performs similar to, and in some cases outperforms, the baseline while providing similar length control capabilities. In particular, our model shines in the most challenging setting: 75-byte sentence lengths.

We also show the results of the ensemble model following ideas of Juraska et al. [8], which provides consistently and significantly better results.

4 Conclusions and Future Work

In this paper, we presented a *modular framework* for controlling length in sentence generation models. As the results show, this framework allows the decoupling of generation and length-control, while yielding performance comparable to that of state-of-the-art models.

Another benefit of modularity is that the structure easily lends itself to ensembling. It is far less expensive to train a stylizer compared to the summarizer so one can quickly obtain an ensemble based on only re-training stylizers while leaving the summarizer fixed. We demonstrated the effectiveness of this technique.

Our work will be extended in two directions. First, we intend to show that our technique can be used with other base summarizers, such as [6]. Second, the stylizer is a module that can be replaced with another model that is trained on a labeled dataset and learns to refine the summarizer outputs in a short period of time due to pre-training. This could be a module that controls tense, sentiment, or style of the sentences. We leave the assessment of other controlled parameters and how they might improve upon our results to future work.

References

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014. URL `http://arxiv.org/abs/1409.3215`.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2016. URL `http://arxiv.org/abs/1409.0473`.

[3] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence RNNs for text summarization. In *Proceedings of Conference of the Association for the Advancement of Artificial Intelligence (CoNLL)*, pages 280–290, 2016.

[4] Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2091–2100, 2017.

[5] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1328–1338, 2016.

[6] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Workshop on Neural Machine Translation and Generation (NMT@ACL)*, 2017.

[7] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[8] Juraj Juraska, Panagiotis Karagiannis, Kevin K. Bowden, and Marilyn A. Walker. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 152–162, 2018.

[9] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100, 2012.

[10] ROUGE: A Package for Automatic Evaluation of summaries. In *ACL Workshop on Text Summarization Branches Out*, page 10, 2004.

[11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, July 2011. ISSN 1532-4435.

[12] Matthew D. Zeiler. ADADELTA: An adaptive learning rate method. ArXiv:1212.5701, 2012.

[13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. Arxiv:1211.5063, 2013.