

Self-Supervised Learning for Contextualized Extractive Summarization

Hong Wang[†], Xin Wang[†], Wenhan Xiong[†],
Mo Yu[‡], Xiaoxiao Guo[‡], Shiyu Chang[‡], William Yang Wang[†]

[†] University of California, Santa Barbara

[‡] IBM Research

{hongwang600, xwang, xwhan, william}@cs.ucsb.edu,
yum@us.ibm.com, {xiaoxiao.guo, shiyu.chang}@ibm.com

Abstract

Existing models for extractive summarization are usually trained from scratch with a cross-entropy loss, which does not explicitly capture the global context at the document level. In this paper, we aim to improve this task by introducing three auxiliary pre-training tasks that learn to capture the document-level context in a self-supervised fashion. Experiments on the widely-used CNN/DM dataset validate the effectiveness of the proposed auxiliary tasks. Furthermore, we show that after pre-training, a clean model with simple building blocks is able to outperform previous state-of-the-art that are carefully designed. ¹

1 Introduction

Extractive summarization aims at shortening the original article while retaining the key information through the way of selection sentences from the original articles. This paradigm has been proven effective by many previous systems (Carbonell and Goldstein, 1998; Mihalcea and Tarau, 2004; McDonald, 2007; Cao et al., 2015). In order to decide whether to choose a particular sentence, the system should have a global view of the document context, e.g., the subject and structure of the document. However, previous works (Nallapati et al., 2017; Al-Sabahi et al., 2018; Zhou et al., 2018; Zhang et al., 2018) usually directly build an end-to-end training system to learn to choose sentences without explicitly modeling the document context, counting on that the system can automatically learn the document-level context.

We argue that it is hard for these end-to-end systems to learn to leverage the document context from scratch due to the challenges of this task, and a well pre-trained embedding model that incorporates document context should help on this

¹Code can be found in this repository: <https://github.com/hongwang600/Summarization>

Masked Paragraph	Last week, I went to attend a one-day meeting. I booked the flight in advanced. [masked sentence] The earliest next flight will be a few days later. I had to use the online discussion instead.
Candidate Sentences	But the flight was cancelled due to the weather. But I lost my passport. The meeting was cancelled. The weather is good today.

Figure 1: An example for the *Mask* pre-training task. A sentence is masked in the original paragraph, and the model is required to predicted the missing sentence from the candidate sentences.

task. In recent years, extensive works (Pennington et al., 2014; Nie and Bansal, 2017; Lin et al., 2017; Peters et al., 2018; Devlin et al., 2018; Subramanian et al., 2018; Cer et al., 2018; Logeswaran and Lee, 2018; Pagliardini et al., 2018) have been done in learning the word or sentence representations, but most of them only use a sentence or a few sentences when learning the representation, and the document context can hardly be included in the representation. Hence, we introduce new pre-training methods that take the whole document into consideration to learn the contextualized sentence representation with self-supervision.

Self-supervised learning (Raina et al., 2007; Doersch et al., 2015; Agrawal et al., 2015; Wang and Gupta, 2015) is a newly emerged paradigm, which aims to learn from the intrinsic structure of the raw data. The general framework is to construct training signals directly from the structured raw data, and use it to train the model. The structure information learned through the process can then be easily transformed and benefit other tasks. Thus self-supervised learning has been widely applied in structured data like text (Okanohara and Tsujii, 2007; Collobert and Weston, 2008; Peters et al., 2018; Devlin et al., 2018; Wu et al., 2019) and images (Doersch et al., 2015; Agrawal et al., 2015; Wang and Gupta, 2015; Lee et al., 2017).

Since documents are well organized and structured, it is intuitive to employ the power of self-supervised learning to learn the intrinsic structure of the document and model the document-level context for the summarization task.

In this paper, we propose three self-supervised tasks (*Mask*, *Replace* and *Switch*), where the model is required to learn the document-level structure and context. The knowledge learned about the document during the pre-training process will be transferred and benefit on the summarization task. Particularly, The *Mask* task randomly masks some sentences and predicts the missing sentence from a candidate pool; The *Replace* task randomly replaces some sentences with sentences from other documents and predicts if a sentence is replaced. The *Switch* task switches some sentences within the same document and predicts if a sentence is switched. An illustrating example is shown in Figure 1, where the model is required to take into account the document context in order to predict the missing sentence. To verify the effectiveness of the proposed methods, we conduct experiments on the CNN/DM dataset (Hermann et al., 2015; Nallapati et al., 2016) based on a hierarchical model. We demonstrate that all of the three pre-training tasks perform better and converge faster than the basic model, one of which even outperforms the state-of-the-art extractive method NEUSUM (Zhou et al., 2018).

The contributions of this work include:

- To the best of our knowledge, we are the first to consider using the whole document to learn contextualized sentence representations with self-supervision and without any human annotations.
- We introduce and experiment with various self-supervised approaches for extractive summarization, one of which achieves the new state-of-the-art results with a basic hierarchical model.
- Benefiting from the self-supervised pre-training, the summarization model is more sample efficient and converges much faster than those trained from scratch.

2 Model and Pre-training Methods

2.1 Basic Model

As shown in Figure 2, our basic model for extractive summarization is mainly composed of two parts: a sentence encoder and a document-level self-attention module. The sentence encoder is a bidirectional LSTM (Hochreiter and Schmidhu-

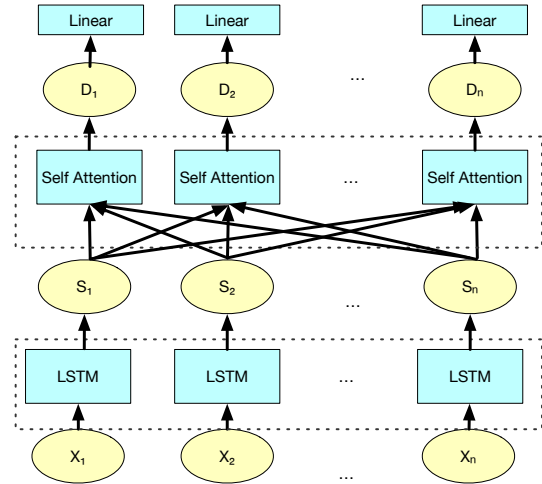


Figure 2: The structure of the Basic Model. We use LSTM and self-attention module to encode the sentence and document respectively. X_i represent the word embedding for sentence i . S_i and D_i represent the independent and document involved sentence embedding for sentence i respectively.

ber, 1997), which encodes each individual sentence X_i (a sequence of words) and whose output vector at the last step is viewed as the sentence representation S_i . Given the representations of all the sentences, a self-attention module (Vaswani et al., 2017) is employed to incorporate document-level context and learn the contextualized sentence representation D_i for each sentence.² Finally, a linear layer is applied to predict whether to choose the sentence to form the summary.

2.2 Self-supervised Pre-training Methods

In this section, we will describe three self-supervised pre-training approaches. Through solving each pre-training task, the model is expected to learn the document-level contextualized sentence embedding model from the raw documents, which will then be used to solve the downstream summarization task. Note that we are only pretraining the sentence encoder and document-level self-attention module of the basic model for extractive summarization.

Mask Similar to the task of predicting missing word, the *Mask* task is to predict the masked sentence from a candidate pool. Specifically, we first mask some sentences within a document with the probability P_m and put these masked sentences ($x_1^m, x_2^m, \dots, x_t^m$) into a candidate pool T^m . The

²We leave the combination of different architectures such as replacing the self-attention module with LSTM for future work.

model is required to predict the correct sentence from the pool for each masked position i . We replace the sentence in the masked position i with a special token $\langle unk \rangle$ and compute its document contextualized sentence embedding D_i . We use the same sentence encoder in the basic model to obtain the sentence embedding S^m for these candidate sentences in T^m . We score each candidate sentence j in T^m by using the cosine similarity:

$$\Theta(i, j) = \cos(D_i, S_j^m)$$

To train the model, we adopt a ranking loss to maximize the margin between the gold sentence and other sentences:

$$\ell_m = \max\{0, \gamma - \Theta(i, j) + \Theta(i, k)\}$$

where γ is a tuned hyper-parameter, j points to the gold sentence in T^m for the masked position i , and k points to another non-target sentence in T^m .

Replace The *Replace* task is to randomly replace some sentences (with probability P_r) in the document with sentences from other documents, and then predict if a sentence is replaced. Particularly, we use sentences from 10,000 randomly chosen documents to form a candidate pool T^r . Each sentence in the document will be replaced with probability P_r by a random sentence in T^r . Let C_r be the set of positions where sentences are replaced. We use a linear layer f_r to predict if the sentence is replaced based on the document embedding D , and minimize the MSE loss:

$$\ell_r = \text{MSE}(f_r(D_i), y_i^r)$$

where $y_i^r = 1$ if $i \in C_r$ (i.e., the sentence in position i has been replaced), otherwise $y_i^r = 0$.

Switch The *Switch* task is similar to the *Replace* task. Instead of filling these selected sentences with sentences out of the document, this task chooses to use sentences within the same document by switching these selected sentences, i.e., each selected sentence will be put in another position within the same document. Let C_s be the set of positions where the sentences are switched. Similarly, we use a linear layer f_s to predict if a sentence is switched and minimize the MSE loss:

$$\ell_s = \text{MSE}(f_s(D_i), y_i^s)$$

where $y_i^s = 1$ if $i \in C_s$, otherwise $y_i^s = 0$.

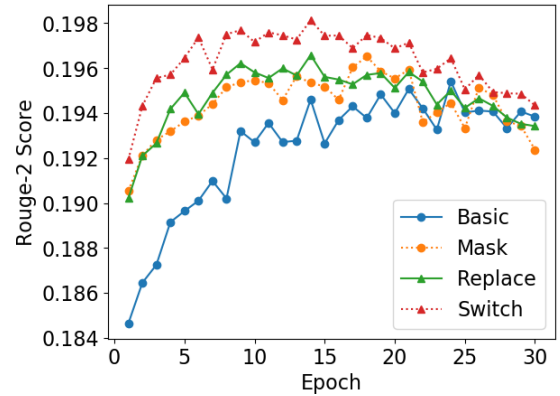


Figure 3: This figure shows the Rouge-2 score for each pre-training method and the basic model on the development set during the training process. We put the result for Rouge-1 and Rouge-L score in Appendix A.2

3 Experiment

To show the effectiveness of the pre-training method (**Mask**, **Replace** and **Switch**), we conduct experiments on the commonly used dataset CNN/DM (Hermann et al., 2015; Nallapati et al., 2016), and compare them with a popular baseline **Lead3** (See et al., 2017), which selects first three sentences as the summary, and the state-of-the-art extractive summarization method **NEUSUM** (Zhou et al., 2018), which jointly scores and selects sentences using pointer network.

3.1 On CNN/DM Dataset

Model and training details We use the rule-based system from (Zhou et al., 2018) to label sentences in a document, e.g., sentences to be extracted will be labeled as 1. Rouge score³ (Lin, 2004) is used to evaluate the performance of the model, and we report Rouge-1, Rouge-2, and Rouge-L as in prior work. We use the pre-trained glove embedding (Pennington et al., 2014) with 100 dimensions to initialize the word embedding. A one-layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) is used as the sentence encoder, and the size of hidden state is 200. A 5-layer Transformer encoder (Vaswani et al., 2017) with 4 heads is used as the document-level self-attention module. A linear classification layer is used to predict whether to choose the sentence.

The training process consists of two phrases. First, we use the pre-training task to pre-train the basic model using the raw article from the

³We use PyRouge <https://pypi.org/project/pyrouge/> to compute the Rouge score.

Method	Rouge-1	Rouge-2	Rouge-L
Basic	41.07	18.95	37.56
LEAD3	39.93	17.62	36.21
NEUSUM	41.18*	18.84	37.61
Mask	41.15*	19.06*	37.65*
Replace	41.21*	19.08*	37.73*
Switch	41.36	19.20	37.86
SentEnc	41.17*	19.04*	37.69*
Switch 0.15	41.35*	19.18*	37.85*
Switch 0.35	41.27*	19.12*	37.77*

Table 1: The Rouge (Lin, 2004) scores for the basic model, baselines, pre-training methods, and analytic experiments. All of our Rouge scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script. The best result is marked in bold, and those that are not significantly worse than the best are marked with *.

CNN/DM dataset without labels. Second, we fine-tune the pre-trained model for the extractive summarization task using the sentence labels. The learning rate is set as 0.0001 in the pre-training phase and 0.00001 in the fine-tune phase. We train each pre-training task until it is converged or the number of training epochs reaches the upper bound 30. We set the probability to mask, replace or switch sentences as 0.25.

Results We show the Rouge score on the development set during the training process in Figure 3, and present the best Rouge score for each method in Table 1. All pre-training methods improve the performance compared with the Basic model. Especially, Switch method achieves the best result on all the three evaluations compared with other pre-training methods, and is even better than the state-of-the-art extractive model NEUSUM⁴.

In the terms of convergence, the Mask, Replace and Switch task takes 21, 24, 17 epochs in the training phase respectively, and 18, 13, 9 epochs to achieve the best performance in the fine-tune phase. The basic model takes 24 epochs to obtain the best result. From Figure 3, we can see that the *Switch* task converges much faster than the basic model. Even adding on the epochs taken in the pre-training phase, Switch method (26 epochs)

⁴We use code from <https://github.com/magic282/NeuSum> to train the model, and evaluate it using our evaluation script. Results using their script (only include Rouge-1 and Rouge-2) is put in Appendix A.1.

takes roughly the same time as the Basic model (24 epochs) to achieve the best performance.

3.2 Ablation Study

Reuse only the sentence encoder Our basic model has mainly two components: a sentence encoder and a document-level self-attention module. The sentence encoder focuses on each sentence, while document-level self-attention module incorporates more document information. To investigate the role of the document-level self-attention module, we only reuse the sentence encoder of the pre-train model, and randomly initialize the document-level self-attention module. The results is shown in Table 1 as SentEnc. We can see that using the whole pre-training model (Switch 0.25) can achieve better performance, which indicates the model learn some useful document-level information from the pre-training task. We notice that only using the sentence encoder also get some improvement over the basic model, which means that the pre-training task may also help to learn the independent sentence representation.

On the sensitivity of hyper-parameter In this part, we investigate the sensitivity of the model to the important hyper-parameter P_w , i.e., the probability to switch sentences. In the previous experiment, we switch sentences with probability 0.25. We further try the probability of 0.15 and 0.35, and show the results in Table 1 as Switch 0.15 and Switch 0.35. We can see Switch 0.15 achieve basically the same result as Switch 0.25, and Switch 0.35 is slightly worse. So the model is not so sensitive to the hyper-parameter of the probability to switch sentences, and probability between 0.15 and 0.25 should be able to work well.

4 Conclusion

In this paper, we propose three self-supervised tasks to force the model to learn about the document context, which will benefit the summarization task. Experiments on the CNN/DM verify that through the way of pre-training on our proposed tasks, the model can perform better and converge faster when learning on the summarization task. Especially, through the Switch pre-training task, the model even outperforms the state-of-the-art method NEUSUM (Zhou et al., 2018). Further analytic experiments show that the document context learned by the document-level self-attention module will benefit the model in summarization

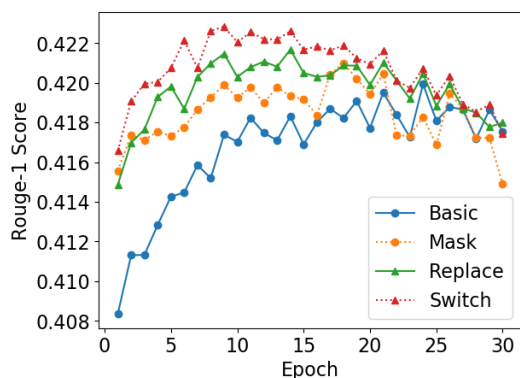
task, and the model is not so sensitive to the hyperparameter of the probability to switch sentences.

References

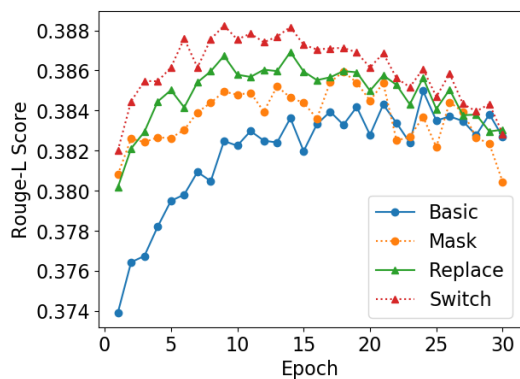
- Pulkit Agrawal, João Carreira, and Jitendra Malik. 2015. [Learning to see by moving](#). In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 37–45. IEEE Computer Society.
- Kamal Al-Sabahi, Zuping Zhang, and Mohammed Nadher. 2018. [A hierarchical structured self-attentive model for extractive document summarization \(HSSAS\)](#). *IEEE Access*, 6:24205–24212.
- Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. [Ranking with recursive neural networks and its application to multi-document summarization](#). In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2153–2159. AAAI Press.
- Jaime G. Carbonell and Jade Goldstein. 1998. [The use of mmr, diversity-based reranking for reordering documents and producing summaries](#). In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 335–336. ACM.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. [Universal sentence encoder for english](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 169–174. Association for Computational Linguistics.
- Ronan Collobert and Jason Weston. 2008. [A unified architecture for natural language processing: deep neural networks with multitask learning](#). In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Carl Doersch, Abhinav Gupta, and Alexei A. Efros. 2015. [Unsupervised visual representation learning by context prediction](#). In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1422–1430. IEEE Computer Society.
- Karl Moritz Hermann, Tomáš Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Computation*, 9(8):1735–1780.
- Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2017. [Unsupervised representation learning by sorting sequences](#). In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 667–676. IEEE Computer Society.
- Chin-Yew Lin. 2004. [Rouge: A package for automatic evaluation of summaries](#). *Text Summarization Branches Out*.
- Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. [A structured self-attentive sentence embedding](#). In *International Conference on Learning Representations*.
- Lajanugen Logeswaran and Honglak Lee. 2018. [An efficient framework for learning sentence representations](#). In *International Conference on Learning Representations*.
- Ryan T. McDonald. 2007. [A study of global inference algorithms in multi-document summarization](#). In *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*, volume 4425 of *Lecture Notes in Computer Science*, pages 557–564. Springer.
- Rada Mihalcea and Paul Tarau. 2004. [TextRank: Bringing order into text](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411. ACL.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. [Summarunner: A recurrent neural network based sequence model for extractive summarization of documents](#). In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3075–3081. AAAI Press.
- Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çağlar Gülçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence rnns and beyond](#). In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

- Yixin Nie and Mohit Bansal. 2017. [Shortcut-stacked sentence encoders for multi-domain inference](#). In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, RepEval@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 41–45. Association for Computational Linguistics.
- Daisuke Okanohara and Jun’ichi Tsujii. 2007. [A discriminative language model with pseudo-negative samples](#). In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.
- Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. [Unsupervised learning of sentence embeddings using compositional n-gram features](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 528–540. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.
- Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. 2007. [Self-taught learning: transfer learning from unlabeled data](#). In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 759–766. ACM.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.
- Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J. Pal. 2018. [Learning general purpose distributed sentence representations via large scale multi-task learning](#). *CoRR*, abs/1804.00079.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.
- Xiaolong Wang and Abhinav Gupta. 2015. [Unsupervised learning of visual representations using videos](#). In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2794–2802. IEEE Computer Society.
- Jiawei Wu, Xin Wang, and William Yang Wang. 2019. [Self-supervised dialogue learning](#). In *ACL 2019, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. The Association for Computational Linguistics.
- Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. [Neural latent extractive document summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 779–784. Association for Computational Linguistics.
- Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. [Neural document summarization by jointly learning to score and select sentences](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 654–663. Association for Computational Linguistics.

A Appendix



(a) Rouge-1



(b) Rouge-L

Figure 4: The Rouge-1 and Rouge-L score for each pre-training method and the basic model on the development set during the training process.

A.1 Evaluation results using scripts from NEUSUM

Method	Rouge-1	Rouge-2
Basic	41.13	18.97
Mask	41.21*	19.07*
Replace	41.27*	19.09*
Switch	41.41	19.22
LEAD3	39.98	17.63
NEUSUM ⁻	41.23*	18.85

Table 2: The Rouge (Lin, 2004) score for basic model, the pre-training methods, and the baselines. We use the script from <https://github.com/magic282/NeuSum> to compute the Rouge score. All of our Rouge scores have a 95% confidence interval of at most ± 0.22 as reported by the official ROUGE script. The best result for each score is marked in bold, and those that are not significantly worse than the best are marked with *.

A.2 Rouge-1 and Rouge-L results

The Rouge-1 and Rouge-L results are shown in Figure 4, from which we can see that the Switch method achieves the best performance.