French Abstractive Text Summarization

Urchade Zaratiana
Johannes Kepler University Linz

Abstract

Most research conducted in the field of Natural Language Processing focuses on the English language. In this paper, we introduce a new dataset for French text summarization. This dataset contains 248k text-summary pairs crawled from Le Parisien and Le Monde websites. We implemented several baselines including Pointer-Generator Networks (See et al., 2017) and transformer-based architecture (Vaswani et al., 2017). Moreover, we show that the usage of the copy mechanism is useful for fast training but it relies too much on extractive summarization.

1 Introduction

Text summarization is the task of reducing the length of a text to a shorter version that captures the main information of the original. Recently, many works have been done in using neural networks to automatically summarize texts (See et al., 2017; Paulus et al., 2018; Liu and Lapata, 2019) all of these works were done for the English language.

In this paper, we introduce a new dataset for French text summarization which contains 248k text-summary pairs obtained by crawling two French news website, Le Parisien and Le Monde.

2 Dataset

In this we section, we present our technique for the extraction of text-summary pairs from two french news websites, Le Parisien and Le Monde. Similar work was done by Nallapati et al. (2016) where they modify a script provided by Hermann et al. (2015) for crawling English text-summary pairs from CNN and Daily Mail. They released the dataset in two versions, a normal and an anonymised version where they replace entity names by special tokens to reduce the size of their vocabulary.

	Number of articles
Le Monde	156396
Le Parisien	91800

Table 1: Number of articles per website.

In our case, we obtained 248190 data pairs in which the source text contain in average 433 words while the summaries consist of 24 words.

3 Experiments

3.1 Preprocessing

We clean the data by stripping accents, removing Unicode characters and HTML tags. We have decided to not put the text to lowercase. Moreover, we filtered out too short summaries and duplicated data.

After that, we trained a Byte-Pair Encoding (BPE) (Sennrich et al., 2015) on our corpora resulting in 30k subwords token. Following See et al. (2017), we truncate the articles to 400 tokens and limit the summaries to 100 tokens. Finally, We use 230k pairs for the training and 9k for both validation and testing.

3.2 Models

LEAD-3 : This model is usually used as a baseline and it consists of selecting the first three sentences as the summary. We also implemented LEAD-1 and LEAD-2 but LEAD-3 performs the best.

PGN and PGN + Coverage : PGN is an LSTM (Hochreiter and Schmidhuber, 1997) encoder-decoder model (Sutskever et al., 2014) with a pointer-generator module introduced by See et al. (2017). The usage of this module improves previous architectures (Rush et al., 2015; Chopra et al.,

Model	R1	R2	R2-F1
LEAD-3	58.50	34.10	16.73
PGN + Coverage	31.43	16.40	16,24
PGN	33.28	17.46	16.74
TransformerABS	12.86	1.44	1.73
LSTM + Attention	17.66	6.31	4.88

Table 2: Performance of the models on the test set.

2016; Nallapati et al., 2016) by allowing the network to copy elements from the source text to handle out-of-vocabulary while retaining the ability to generate new words for the summary. Our model has a 512-dimensional hidden size for both encoder and decoder, a 256-dimensional word embeddings initialized randomly. We apply additive attention (Bahdanau et al., 2014) to allow the model to focus only on information relevant to the generation of the next target word during decoding. We have implemented two variant of this architecture, with and without coverage mechanism. The coverage mechanism is an additional term that helps to avoid repetitions during decoding. We refer readers to (See et al., 2017) for further details.

LSTM + Attention : This model is a simple LSTM encoder-decoder with attention mechanism (Bahdanau et al., 2014). We employ the same hidden size and word embedding dimension as the previous models to all allow fair comparison between them.

TransformerABS : We also implement a transformer (Vaswani et al., 2017) baseline with 6-layers encoder and 6-layer decoders. Following, (Liu and Lapata, 2019) we used 768 hidden size and 2,048 feed-forward filter size. This model has more parameters so the result is not comparable.

3.3 Training

All the models were trained with a batch size of 16 during 4000 steps. We couldn't train longer due to memory constraint. An adam (Kingma and Ba, 2014) optimizer with 0.001 learning rate were used to minimize the Negative Log-Likelihood Loss. Every 1000 steps, we decay the learning rate by 0.5 and we save the model states. During testing, we load the state in which the model got the highest validation performance. All the implementation were done with PyTorch (Paszke et al., 2019) and OpenNMT (Klein et al., 2017).

3.4 Summary Generation

To generate summaries, we employed beam search with a beam size of 5 because we find that greedy decoding yield to low-quality result. Moreover, we used trigram blocking to prevent repetitions during decoding which is critical for generating good quality summaries.

3.5 Results

Table 2 summarize the results for each of the models. We can see that LEAD-3 obtained the best performance for two metrics, the ROUGE-1 and ROUGE-2. For abstractive summarization, the Pointer-Generator Network without coverage outperforms the other baselines.

The transformer model and the LSTM with attention under-perform with large margin the PGN-based models while being trained with the same amount of steps. Its shows that the copy mechanism is very important for faster training as noticed (See et al., 2017).

Figure 1 shows the summary generation with the implemented models. Actually, all of the models produce good summaries for this particular example. We can see that we got more abstractive summaries with TransformerABS and LSTM-attention whereas PGN-based model's summaries are almost extractive.

4 Conclusion and future works

We have presented a novel dataset for french text summarization and have implemented some baseline models. We have seen that pointer-generator networks can reach good performance with only a few training steps.

Furthermore, a better result could be reached by training for more steps and utilizing a larger batch size because the model was still underfitting when we stopped training.

Future work will focus on applying Pretrained Language Models like BERT (Devlin et al., 2018) for both extractive and abstractive text summarization on this dataset. We also looking forward to collecting data for more languages and using them to develop a multilingual text summarization.

Text (Truncated): C' est ce qu' on appelle une entrée fracassante. Ce jeudi, alors que les 28 chefs d' État et de gouvernement se sont retrouvés en début d' après-midi à Bruxelles (Belgique) pour un Conseil européen dédié à la question des migrants en Europe, le Premier ministre italien a d' emblée mis un coup de pression. Nous attendons des actes , a tempêté Giuseppe Conte, avant de prévenir ses homologues : faute de preuves de solidarité de la part des autres pays de l' Union européenne, le chef du nouveau gouvernement populiste bloquerait l' adoption d' un texte commun. Jeudi soir, il a mis sa menace à exécution en attendant de savoir s' il obtiendrait satisfaction sur ses exigences dans le dossier migratoire. Une manière de faire capoter ce sommet, après plus de deux semaines de bras de fer diplomatiques autour du sort des navires humanitaires.
Ground Truth: Les divergences entre les 28 sur la question des migrants sont si fortes qu' il sera difficile de parvenir à un texte commun à l' issue du Conseil européen.
Pointor-Generator + Coverage: Les 28 chefs d' État et de gouvernement se sont retrouvés en début d' après-midi à Bruxelles (Belgique) pour un Conseil européen dédié à la question des migrants de l' Union européenne.
Pointor-Generator: Le chef du nouveau gouvernement populiste bloquerait l' adoption d' un Conseil européen dédié à la question des migrants en Europe.
LSTM + Attention: Pour la première fois , l' Union européenne s' est imposé à l' issue de la réforme des droits de l' opposition.
TransformerABS: Le chef de l' Etat a annoncé qu' il n' y a pas d' accord avec les Etats-Unis.

Figure 1: Summary generation with the models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. 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*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *CoRR*, abs/1701.02810.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP/IJCNLP*.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. *ArXiv*, abs/1705.04304.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *ArXiv*, abs/1704.04368.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.