Résumé Résumant Les Nouvelles

Abstractive News Summarization in French

Name	USC email	USC ID
Melvin Mathew	melvinm@usc.edu	3464393508
Sarthak Arora	sarthaka@usc.edu	2455179070
Mun Kit Teh	mteh@usc.edu	5959252603
Prakarsh Upmanyu	upmanyu@usc.edu	4686839584

Github repository
https://github.com/silicon-beach/news-in-short

Division of labor
Research: Melvin, Mun Kit
Corpus Creation: Prakarsh, Sarthak
Algorithm Design and Implementation: Melvin, Prakarsh, Mun Kit, Sarthak
Report: Melvin, Prakarsh, Mun Kit, Sarthak

Word count
2043 words

INTRODUCTION

Every day people rely on a wide variety of news sources to gather and comprehend information as well as make informed decisions about pertinent issues. But, the limitation of time caters to the reluctance of many individuals to read long-winded news articles. As in most cases, the headlines are often misleading and insufficient, warranting a need for a synoptic view of an article.

Traditional text summarization techniques have mostly relied on identifying and extracting key text fragments like words or sentences using word-level or sentence level statistical analysis and human-engineered features over the text corpus. Since extractive summarization is just based on statistical features and not the semantic relationship between words or sentences, these summaries often turn out to be inconsistent.

Motivated by the aforementioned reasons, we propose an application that encapsulates the daily news in French in 50 words. We use an encoder-decoder model with LSTM units to generate abstractive summaries from the text of multiple political news articles. Abstractive summarization is an elusive technological capability in which textual summaries of content are generated de novo (Fei Liu et al., 2015). The vanilla encoder-decoder sequence to sequence learning model using LSTMs as described in Sutskever et al., 2014 is shown in Fig 1

Although many deep learning methods have been proposed to summarize news articles in English, fewer strategies exist to concisely and coherently paraphrase news articles in French. In the end, we also go on to propose the use of a relatively recent and novel evaluation mechanism called FRESA for automatic evaluation of textual summaries, alleviating the problem of procuring reference summaries. This is because FRESA works by assigning a score based on how much content from the original article is being reflected in the generated summary unlike other evaluation metrics such as Rouge which require reference summaries. Subsequently, in the last section, we conclude by discussing threats to validity and future work.

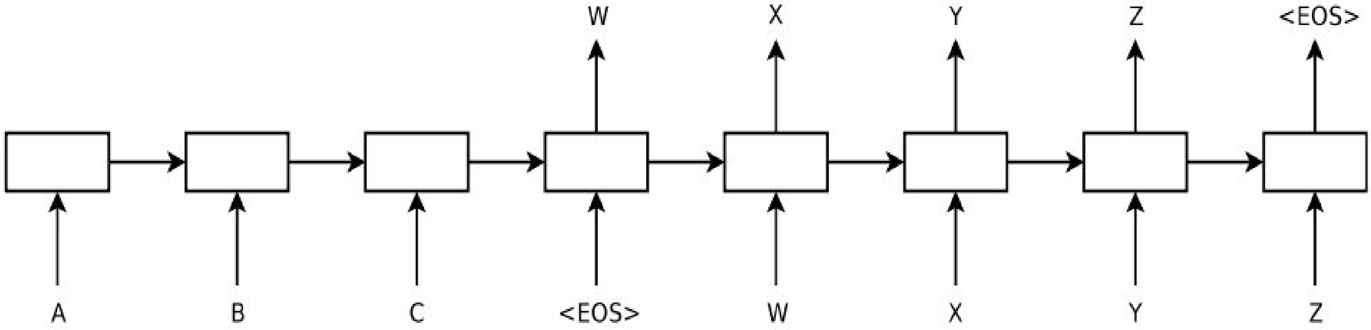

Figure 1: The model takes as input a sequence X = (A,B,C) ending with a end-of-sentence token and produces an output sequence Y=(W,X,Y,Z). The end-of-sentence symbol at the end signals the decoder to stop producing the output.

Method

Materials

For effective summarization, our proposed model requires lots of training data. Therefore, we collected a total of 500,000 french political news articles from various online news sites. In order to achieve this, we created a web crawler using Scrapy - a python based web crawling framework. Since different news sites have different layouts and page structure, we created a base crawler and constantly modified it to correctly detect URLs of different news articles on different news sites.

Some of the websites that we crawled are as follows:

- www.lesechos.fr
- www.leparisien.fr
- www.lexpress.fr
- www.humanite.fr
- www.lemonde.fr
- www.ouest-france.fr
- www.huffingtonpost.fr
- www.francesoir.fr
- www.latribune.fr
- www.lalsace.fr
- www.20minutes.fr
- www.charentelibre.fr
- www.ladepeche.fr
- www.lopinion.fr
- www.ledevoir.com

Subsequently, the URLs obtained are fed into a generalized scraper that is implemented using python-goose - an article extraction library. Since it is a generalized scraper, it is capable of extracting headlines and articles from each of the crawled URLs regardless of page structure. The data was then cleaned and pre-processed to handle irregularities in case of missing article or headline texts.

Listed below are details of the scraped data:

- Quantity: 446,000 (headline, article) pairs
- 7,496,230 words
- The data can be downloaded from https://s3-us-west-1.amazonaws.com/news-in-short/data.zip

Procedure

For this task we use a simple sequence to sequence learning model which consists of 2 recurrent neural networks with long short term memory (LSTM) units. The LSTM's ability to successfully learn on data with long range temporal dependencies and mitigate the problem of vanishing gradients makes them an obvious choice for the task of summarization. The first LSTM network is an encoder that takes as input a news article to build a large fixed-dimensional vector representation of the input article. The second LSTM network is a decoder that takes as input the encoder's output and generates a summary of up to 50 words. We then use beam search to select the most suitable summary out of all candidate summaries generated by the model.

In order to achieve this, we implemented 3 main components:

1. Word2vec embedding generator
2. Trainer
3. Predictor

Word2vec embedding generator

Understanding the meaning of a word is at the center of any natural language processing task. While a deep, human-like, understanding remains elusive, many methods have been successful in capturing certain aspects of similarity between words (Levy et al., 2015). Therefore, to capture the semantic meaning of most frequent words in our corpus we trained our corpus using word2vec to obtain a high dimensional vector representation of the words. Word2vec uses a simple feed-forward neural network to learn a probabilistic language model over the input using an easily scalable noise contrastive estimation loss.

Trainer

The learned embeddings are used to initialize the first layer weights of the encoder LSTM network. These weights are then further fine-tuned using the article words as the input and the corresponding headline words as the labels using a simple cross entropy loss measure. This helps the encoder LSTM network to learn a fixed-dimensional representation over the input article.

Predictor

The decoder uses the same LSTM architecture as the encoder. In order to generate effective summaries using LSTM units, we use the concept of attention mechanism which helps the decoder to decide over the importance of words while generating the output summary word by word.

We optimize the model using RMSprop, minimizing the cross-entropy loss, by backpropagating through the network:

$$CrossEntropy(y,\widehat{y}) \ = \ -\sum_{i=1}^{|V|} y_i \, log(\, y_i \,) \qquad (1)$$

where y is the target sequence and $\widehat{y}$ is the softmax output at the decoder.

Evaluation

In order to evaluate the performance of the summarizer, we use the Framework for Evaluating Summaries Automatically (FRESA). We chose FRESA because it is capable of evaluating generated summaries without the need for reference summaries. Reference summaries in French are not easy to find. FRESA is similar to ROUGE and works by calculating the divergence between the generated summary and the original article. In this case, we will be using an implementation based on the Kullback-Leibler (KL) and Jensen-Shanon (JS) divergence.

FRESA calculates the following scores:

1. FRESA_1 (unigrams)
2. FRESA_2 (bigrams)
3. FRESA_4 (skipgrams)
4. FRESA_M (average of FRESA_1, 2, 4)

Each of the scores listed above are normalized between 0 and 1. In regards to metrics, a high FRESA score would signify low divergence from the original article. This also means that a higher FRESA score would indicate that more information from the source article is being reflected in the generated summary.

In regards to how the evaluation is performed, we first calculated the FRESA score of a small subset of summaries generated by other summarizers and use them to determine baseline scores. Subsequently, we compare the FRESA score for the summaries generated by the proposed summarizer with the baseline score. Since a higher FRESA score indicates a better summary, the proposed summarizer should at least obtain a FRESA_M score similar to or better than the baseline score.

We chose 100 articles of similar lengths and calculated the average of their FRESA scores. Since the FRESA score reflects presence of content from the source article, it is only fair if similar length articles are chosen as article length is somewhat proportional to amount of content. Initially, we attempted to search for publicly available abstractive summarizers but were unable to find any. Therefore, we resorted to procuring extractive summaries generated by summarizers such as Artex, Cortex and Enertex. We then proceeded to average the scores obtained by each of the summarizers and used them to establish baseline scores. The results are there in the ‘Results’ section.

Results

We implemented the proposed summarizer as a 3 layer RNN with 512 nodes in each layer. We trained the model with a batch size of 32 using RMSprop optimizer. Due to the need to train the RNN on large volumes of training data, we utilized Google Cloud for its computing capacity.

Listed below is our Google Cloud configuration:

1. GPU - 10
2. CPU - 80
3. RAM - 80 * 6.5 GB

When we first trained the RNN using all the data that we scraped, we ran into memory issues. Therefore, we decided to train the RNN using approximately 3/4th the amount of scraped articles as anything more would be too computationally expensive. The trainer RNN took a total of approximately 28 hours in order to finish training.

Shown below are the average FRESA scores obtained:

Summarizer	**FRESA_1**	**FRESA_2**	**FRESA_4**	**FRESA_M**
News-in-short (proposed)	**0.21104**	**0.09057**	**0.08697**	**0.12953**
Artex	0.23693	0.11841	0.11454	0.15662
Cortex	0.2859	0.13942	0.13134	0.18493
Enertex	0.30203	0.13842	0.13134	0.19093

Table 1: FRESA scores comparison between the proposed summarizes with baseline summarizers.

As shown in the table above, the proposed summarizer failed to obtain scores higher than the other summarizers. Despite that, we consider our scores acceptable as quality abstractive summarization is a difficult task to accomplish. We also believe that our scores are partly attributed to the fact that we had limited computing resources to train the system on more data.

Following is a sample translation of the original text:

- Les trois pays qui ont signé l'Accord de libre-échange nord-américain (ALENA) Alena, les États-Unis, le Mexique et le Canada, étaient prêts à renégocier cet accord commercial rapidement … (original language)
- The three country who have sign the agreement of free exchange North American (NAFTA) Alena, the United States, the Mexico and the Canada, were loans at renegotiate this agreement commercial quickly … (a word-by-word gloss)
- The three countries that signed the North American Free Trade Agreement (NAFTA) Alena, the United States, Mexico and Canada, were ready to renegotiate this trade agreement quickly… (english translation)

Following is a sample output of the proposed summarizer:

- Trois pays porteurs répondent, présentent, négocient le commerce rapidement révèle un vote équivalent jugé rapidement, date de négociation … (original language)
- Three country bearers respond, have, negotiate the trade quickly reveals a vote equivalent judge quickly, dated of negotiation … (a word-by-word gloss)

- Three country bearers reply, present, negotiate trade fast reveals vote equivalent judged quickly, date of negotiation … (english translation)

Discussion

In this work we attempted to build a deep neural network based approach to summarization targeting the particular domain of political articles. A vast majority of the previous approaches to summarization have been extractive, where sentences and key phrases are ranked using statistical measures like word frequency measure, position in the text and lexical-chains. On the contrary, we proposed an attention based neural network model deriving from its recent success in machine translation system which operate on sequences of data.

Most of the deep learning approaches to text summarization use large amount of articles to achieve state-of-the-art results. However, due to a limitation in computational resources we could only train the model on about 3/4th the amount of scraped articles. We believe an increase in the dataset size and the number of GPU instances will further help us improve our FRESA scores.

Furthermore, we used a simple LSTM based encoder-decoder model which could compute attention weights for only the current input word given the previous word. However, using a bidirectional RNN model we can compute attention weights for the words following the current word, thereby giving more information to the decoder while generating sentences. Moreover, getting state-of-the-art results on a corpus of multi-domain documents poses a challenge for future implementations.

One more point to consider here is that since our training data consisted of mainly French political news, the resulting summaries, from the proposed summarizer, of general articles does not go as well as one would like it to be. The summaries are little biased towards France and uses words from political domain incorrectly. In order to improve upon this shortcoming, we can generate training data from multiple domains from around the world.

In closing these results suggest that by either using a more robust seq2seq architecture or by training on a larger dataset our model will produce more consistent and meaningful summaries with even better FRESA scores.

References

[1] Liu, F., Flanigan, J., Thomson, S., Sadeh, N., & Smith, N. A. (2015). Toward abstractive summarization using semantic representations.

[2] Saggion, H., Torres-Moreno, J. M., Cunha, I. D., & SanJuan, E. (2010, August). Multilingual summarization evaluation without human models. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 1059-1067)*. Association for Computational Linguistics.

[3] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *In Advances in neural information processing systems* (pp. 3104-3112).

[4] Lopyrev, K. (2015). Generating news headlines with recurrent neural networks. *arXiv preprint arXiv*:1512.01712.

[5] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

[6] Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, *3*, 211-225.

[7] Torres-Moreno, J. M. (2012). Artex is another text summarizer. *arXiv preprint arXiv:1210.3312*.