Multilingual Automated Summarization

Anjali Agrawal aa7513@nyu.edu
Prashant Garmella pg1910@nyu.edu
Jeewon Ha jh6926@nyu.edu
Anthony Lanzisera acl673@nyu.edu

Abstract

We aim to develop an automated system for multi-lingual text summarization to effectively capture important features of the text in various languages. In this project, we propose the pipeline for the task (Source language: Spanish; Target language: English) as a two-stage approach - translating the text from Spanish to English and then summarizing the English article. We utilize OpenNMT to train and evaluate our models. We leverage various parallel corpora to verify our hypothesis that spoken/informal language corpus helps translation models have better generalizability than the formal datasets do for domain-specific applications. The evaluation results show that the models trained on the spoken language dataset perform better compared to the models trained on the formal language dataset.

1 Introduction

Text summarization gives a condensed version of a detailed article, including the main ideas and features of a text. Text summarization divides into two branches: extractive and abstractive. Extractive summarization keeps extracted sentences' objects unchanged, whereas abstractive summarization involves paraphrasing (Khatri et al., 2018). Since the advent of globalization, it has been increasingly important for sharing summarized news and information in multiple languages. News articles get stale very quickly, so having a near real-time summarization and translation system for articles would benefit many. Furthermore, even though more computing power is available, we still need quicker NLP systems to compete. These inspire us towards the idea of multilingual text summarization (Scialom et al., 2020). Multilingual text summarization can play a pivotal role in global news sharing, and in sharing knowledge and information in multinational companies, global education institutions, and military units. We intend to implement and improve the way to generate a text summary for a document in a language other than the original document language. We also aim to evaluate whether training a machine-translation model from domain-specific data could yield better translations for a given application or not. In this project, we have translated articles from Spanish to English and then summarized the translated articles using OpenNMT alongside the most current best practices in NLP. We perform translation prior to the summarization because it could be easily and efficiently extended to under-resourced languages where it is difficult to find article-summary pair corpora in the source language.

2 Related Work

Until MLSUM arrived, the research area of Multilingual text summarization had focused on extractive processing (models can include Oracle, Random, Lead-3, and TextRank) since no sufficient training corpora were ready (Scialom et al., 2020). MEAD was an early pioneer, translating between English and Chinese (Scialom et al., 2020). MultiLing covered 40 languages, but it lacked examples (Scialom et al., 2020). Abstractive summarization techniques include encoder-decoder neural architectures, such as Pointer-Generator (Scialom et al., 2020). Also, M-BERT, or Multilingual Bert, is a popular way of pretraining the summarization task, whereby models can be fine-tuned to leverage previously gained information picked up from pretraining on large corpora (Scialom et al., 2020). MLSUM was the first large-scale "MultiLingual SUMmarization" dataset (Scialom et al., 2020). It originated from online newspapers and is a 1.5 million article-to-summary parallel corpus. (Rush et al., 2015) used local attention for an input sentence generated summaries using only data to fuel their results. (Pontes et al., 2018) used chunks and

compression methods, called Multi-Sentence Compression (MSC). OpenNMT's attention mechanism has been extended to include local, sparse-max, hierarchical, and structured attention (Klein et al., 2020). The work in (Zhu et al., 2020) also incorporates machine translation and mono-lingual summarization techniques to develop a neural cross-lingual summarization system for Chinese to English.

3 Methodology

In this section, we propose a two-stage process for obtaining the English summary from a Spanish text: 1) translation model returns the translated article as an intermediate output, 2) summarization model intakes the output from the first step to produce summaries[1]. Figure 1 depicts the overall approach for our objective. An example of the input and the output from the pipeline is highlighted in Appendix A.

Figure 1: Illustration of the Overall Approach

3.1 OpenNMT

OpenNMT is an open-source project supporting diverse architectures that can be deployed for neural machine translation and related tasks. For both tasks, we used OpenNMT-py, which is a "user-friendly and multimodal implementation benefiting from PyTorch ease of use and versatility." (Klein et al., 2020)

3.2 Data

We used Europarl corpus (Koehn, 2011) and TED Talks corpus (Tiedemann, 2012) for training and evaluating translation models. Europarl is a parallel corpus extracted from the proceedings of the European Parliament; it consists of 2M sentences. TED Talks is another parallel corpus, consisted of 1.6M transcribed and/or translated sentences extracted from TED website. We filtered out any sentences that are space/punctuation only. From each corpus, we selected 100,000 parallel sentences and randomly split them into 80,000 for training, 10,000 validation, and 10,000 test. All datasets were processed using the standard preprocessing techniques available through OpenNMT-py (Klein et al., 2017).

We utilize the CNN-Daily Mail dataset (Hermann et al., 2015) and Gigaword dataset (Rush et al., 2015) for training and evaluating summarization models. CNN-Daily Mail dataset contains 3M article-summary pairs. The training, validation, and test datasets contain 287,226, 13,368, and 11,490 article-summary pairs, respectively. On average, each article has 781 tokens, while its corresponding summary contains 56 tokens. The Gigaword dataset contains 4M article-summary pairs. The training, validation, and test datasets contain 3,803,957, 189,651, 1,951 headline-summary pairs, respectively. On average, each headline has 26 tokens, while its corresponding summary 8 tokens. All datasets were processed using the standard preprocessing techniques available through OpenNMT-py (Klein et al., 2017).

4 Experimental Setup

4.1 Implementation

The experiments were performed using the computing resources available through Google Colaboratory. The implementation was done in Python, using the functionalities of Pytorch and OpenNMT-py. We also utilize Python libraries for analysis and evaluation.

4.1.1 Translation

For the Spanish-to-English translation task, we propose a 3-step process. First, we plan to obtain baseline performance of the translation models trained on a single corpus, using OpenNMT with its basic configuration. Then, we will perform hyperparameter tuning to identify the optimal configuration for training each corpus. All of the models were trained from scratch, and the hyperparameters used for fine-tuning are the same as the default settings provided by OpenNMT, except for RNN type, optimizer, learning rate, learning rate decay, and training steps. Lastly, we will examine the models' generalizability on domain-specific data by evaluating them on the test set, which contains a balanced representation of the sentences from both the Europarl and Ted Talks corpora.

4.1.2 Summarization

For summarization, we experiment with models trained using the two datasets - CNN Daily Mail, and Gigaword. For the CNN Daily Mail corpus,

[1]The current approach is developed only for Spanish to English, however, with the available parallel corpora, we can easily train similar models to include more languages.

Model	Validation Accuracy	Validation Perplexity	BLEU	ROUGE-L (F1)
EP_baseline	48.028	32.736	0.563	0.455
EP_gru_sgd	43.371	34.769	0.552	0.443
EP_lstm_adadelta	46.900	**32.162**	0.563	**0.458**
EP_lstm_adam	45.687	111.359	0.550	0.432
EP_lstm_decay	**48.113**	32.641	**0.568**	0.456
TT_baseline	**51.134**	31.925	0.571	**0.479**
TT_gru_sgd	48.077	32.381	0.542	0.450
TT_lstm_adadelta	50.262	**27.605**	**0.575**	0.473
TT_lstm_adam	49.283	206.934	0.558	0.463
TT_lstm_decay	50.973	32.299	**0.575**	**0.479**

Table 1: Translation (Spanish to English) Results for various models for v

Model	BLEU	ROUGE-1 (F1)	ROUGE-2 (F1)	ROUGE-L (F1)
CNN-DM_batch_1 (1-layer Bi-LSTM)	**53.0**	**43.0**	**20.1**	**39.0**
CNN-DM_batch_16 (1-layer Bi-LSTM)	49.0	42.3	20.0	38.9
CNN-DM_batch_16 (Transformers)	44.0	42.3	20.0	38.8
Gigaword_copy_attention (2-layer LSTM)	**36.0**	**35.7**	**17.2**	**34.6**

Table 2: Summarization (English summaries) Results for various models

due to limited computing resources and the size of the training data, we use different pre-trained models, which are trained on the CNN Daily Mail dataset, and evaluated the models with different inference parameters to obtain the best performing model. For Gigaword, we train a model (2-layer LSTM with copy attention) from scratch using the default configuration provided by OpenNMT-py and evaluate it.

4.2 Evaluation

Similar to the work presented in (Klein et al., 2017), we used BLEU (Bilingual Language Understudy) as well as ROUGE (the Recall-Oriented Understudy for Gisting Evaluation) scores for evaluating the models. BLEU is a precision-based metric, which measures the proportion of n-grams in the prediction which also appears in the golden. Vice versa, ROUGE is a recall-based metric, which measures the proportion of n-grams in the golden which also appears in the prediction. Under the consideration that both properties have to be maintained to construct good translation and summarization models, we incorporated both metrics to compare our results with the baseline models.

4.3 Results and Analysis

This section outlines the results for translation as well as summarization and highlights some of our observations.

4.3.1 Translation

The results for Translation are illustrated in Table 1; for a detailed configuration of each model, see Appendix B. Based on the reported performance, we found that the best performing translation models were **EP_lstm_adadelta** for Europarl and **TT_lstm_decay** for TED Talks. For generalizability test using the combined dataset, while the model trained on Europarl achieved BLEU score of 0.501 and ROUGE-L score of 0.390, the model trained on TED Talks achieved BLEU score of 0.516 and ROUGE-L score of 0.414. The results show that the model trained on TED Talks achieved better performance than the model trained on Europarl which supports our hypothesis that the translation model trained on the informal text will learn better representations than the model trained on the formal text.

4.3.2 Summarization

Table 2 outlines the results for summarization. We observe that among the models trained on the CNN Daily Mail, the Bi-LSTM model with the basic configuration outperforms the other models in terms of BLEU and ROUGE scores[2]. Therefore, we choose the 1-layer Bi-LSTM model with the default configuration to be the best performing model trained on CNN Daily Mail. For the model trained on Gigaword, we achieve the results similar to those produced by the models available through OpenNMT-py (Klein et al., 2017). We use the best performing model trained on CNN Daily Mail for our pipeline, since extracting highlights would be more beneficial for news articles when determining the important aspects of the text. We can easily interchange the model with the one trained on Gigaword if we

[2]Due to GPU availability limit on Google Colaboratory, we were unable to try more configurations, therefore we select the best model from the available ones.

want to extract the headlines of the article.

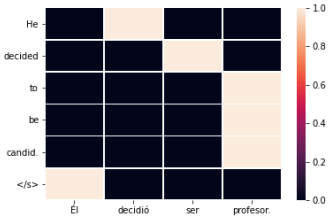

Figure 2: Attention maps for the translation model trained with Europarl data

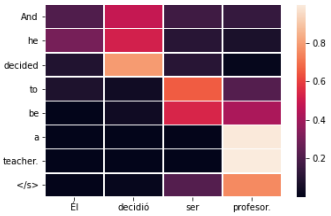

Figure 3: Attention maps for the translation model trained with Ted-Talk data

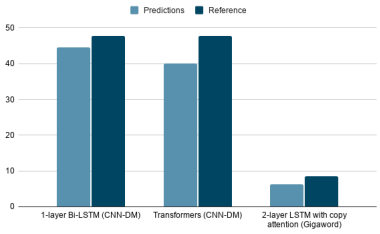

Figure 4: Average Number of Unique Tokens generated in Abstractive Summarization of the datasets

4.4 Analysis

To interpret the outputs of the translation models, we look into the attention maps, which highlight the focus of the network at each time step. Figure 2 and Figure 3 shows the attention maps for a sample Spanish sentence that is translated into English using the best models for the Europarl (EP_lstm_adadelta) and TEDTalks (TT_lstm_decay), respectively. We can observe that the model trained with TedTalks produces better translation by delicately identifying important features to focus on at each time step. The translation generated by EP_lstm_adadelta is not only correctly translates 'professor' to 'a teacher' but also captures the singularity of the word 'teacher' even in the absence of 'un'. The result aligns with our hypothesis that informal language data benefit the translation model in case of domain-specific application.

For the task of abstractive summarization, having more unique tokens might be beneficial since it would not only extract the parts of passage but also be generating words to write the abstract/summary. Analyzing the number of unique tokens enables us to understand the new words that the model utilizes while generating the summary in comparison to the reference/human-written summaries. Figure 4 shows the number of unique tokens used for generating summaries by the models trained using CNN Daily Mail Data and Gigaword data in comparison to the reference summaries. We see, for the models trained using CNN Daily Mail data, that the model with higher BLEU and ROUGE scores (1-layer Bi-LSTM) also have more number of unique tokens in their generated summaries. The model trained using the Gigaword dataset also uses unique tokens near the same range to the reference summaries. The number of tokens are less in comparison to the models trained using CNN Daily Mail datatset is because the the Gigaword dataset contains the headlines as the summary in contrast to the manual highlights available for the CNN-DM dataset.

5 Conclusion & Future Work

In this project, we propose an indirect pipeline for multilingual automated summarization - we first translate from Spanish article to English article, then generate the summary for the translated text. Based on the results, we confirmed our hypothesis that the translation model trained on TED Talks corpus (informal language) has better generalizability in comparison to the translation model trained on Europarl corpus (formal language) for domain-specific application.

Multilingual automated summary generation (direct pipeline) may effectively extract important features from the original text and achieve better performance by utilizing information from the source language as well as the target language. As a part of our future work, both translation and summarization models could be further improved by searching a larger hyperparameter space and training the models using high computing resources. Moreover, we would like to 1) automate and standardize the ETL pipeline for different input formats, 2) incorporate more diverse language pairs, and 3) compare the performance between indirect and direct (Spanish text to English summarization) multilingual text summarization.

6 Collaboration Statement

Anthony Lanzisera (acl673): Team idea generation, TED Talks translation model generation, Related works.

Anjali Agrawal (aa7513): Team idea generation, CNN Daily Mail summarization model generation, Methodology, Result & Analysis.

Jeewon Ha (jh6926): Team idea generation, Europarl translation model generation, Methodology, Result & Analysis.

Prashant Garmella (pg1910): Team idea generation, Gigaword summarization model generation, Conclusion & Future work.

References

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.

Chandra Khatri, Gyanit Singh, and Nish Parikh. 2018. Abstractive and extractive text summarization using document context vector and recurrent neural networks. *arXiv preprint arXiv:1807.08000*.

Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The OpenNMT neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 102–109, Virtual. Association for Machine Translation in the Americas.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Philipp Koehn. 2011. Europarl: A parallel corpus for statistical machine translation. Citeseer.

Elvys Linhares Pontes, Stéphane Huet, and Juan-Manuel Torres-Moreno. 2018. A multilingual study of compressive cross-language text summarization.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. Mlsum: The multilingual summarization corpus.

Jrg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Junnan Zhu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2020. Attend, translate and summarize: An efficient method for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1309–1321, Online. Association for Computational Linguistics.

A Example of Generating the Summary using the Automated Pipeline

Consider a sample Spanish article[3]-

Sample text (ES): "La pequeña ciudad texana, cercana a la frontera con México, tiene un centro comercial al borde del camino y el obligatorio Starbucks pero, a primera vista, poco más. Sin embargo Xochitl Mora insistió....."

Sample text (EN): "The small Texan city, not far from the border with Mexico, has a roadside shopping mall and an obligatory drive-thru Starbucks - but at first glance, not much else. But Xochitl Mora was insistent..." (**1133 tokens**)

Ted talk Translation: "At the end end close to the border with Mexico, the border in Mexico, has a shopping mall on the edge of the way and the impaired. However, they Xochitl insistió....." (**BLEU**: 0.432, **ROUGE-L**: 0.302)

Europarl Translation: "My small city is in the way to see the border with Mexico, which has a central role to the road and compulsory compulsory compulsory but, at the first time, more than more. Yet it does not!..." (**BLEU**: 0.35, **ROUGE-L**: 0.236)

Ted talk Translation + CNN Summarization: "the mayor of his McAllen, Jim Spears went to south after the war for not to be able to strike the algae, the inviernos from his nativo New York City.For . The conductor of the harbor told me, "Don't give me 15 niños'.We're not accustomed to dealing with niños", it's reminiscent of Darling."In issue of two days, we had a snapshot diferente".The who the mayor were rehúsa to call a crisis, was the sudden de of the gran." (**81 tokens**)

[3]Source: https://www.bbc.com/mundo/noticias-37827365

Here, we can observe that the model trained using Ted-Talk data produces better translations than the translations produced using the model trained using Europarl dataset. We also observe that the summarization model gives the key ideas of the articles but are not in par with human summaries since they can be easily distinguished as machine-generated due to some grammatical errors. Therefore, improving the summarization models would enable us to get better summaries for the articles.

B Configurations used for fine-tuning translation models

- data_baseline: rnn type (LSTM), optimizer (SGD), learning rate (1), learning rate decay (0.5), and training steps (50000)
- data_gru_sgd: rnn type (GRU), optimizer (SGD), learning rate (1), learning rate decay (0.5), and training steps (50000)
- data_lstm_adadelta: rnn type (LSTM), optimizer (Adadelta), learning rate (1), learning rate decay (0.5), and training steps (50000)
- data_lstm_adam: rnn type (LSTM), optimizer (Adam), learning rate (0.001), learning rate decay (0.5), and training steps (50000)
- data_lstm_decay: rnn type (LSTM), optimizer (SGD), learning rate (1), learning rate decay (0.1), and training steps (50000)