Proceedings of the 4th Financial Narrative Processing Workshop @ LREC 2022, pages 59–64
Marseille, 24 June 2022

Extractive and Abstractive Summarization Methods for Financial Narrative Summarization in English, Spanish and Greek

Alejandro Vaca, Alba Segurado, David Betancur, Álvaro Barbero*
IIC (Instituto de Ingeniería del Conocimiento)
* Universidad Autónoma de Madrid
{alejandro.vaca, alba.segurado, david.betancur, alvaro.barbero}@iic.uam.es

Abstract

This paper describes the three summarization systems submitted to the Financial Narrative Summarization Shared Task (FNS-2022). We developed a task-specific extractive summarization method for the reports in English. It was based on a sequence classification task whose objective was to find the sentence where the summary begins. On the other hand, since the summaries for the reports in Spanish and Greek were not extractive, we used an abstractive strategy for each of the languages. In particular, we created a new Encoder-Decoder architecture in Spanish, MariMari, based on an existing Encoding-only model; we also trained multilingual Encoder-Decoder models for this task. Finally, the summaries for the reports in Greek were obtained with a translation-summary-translation system in which the reports were translated to English and summarised, and then the summaries were translated back to Greek.

Keywords: Extractive Summarization, Abstractive Summarization, Multilingual Models, Encoder-Decoder

1. Introduction

Given the increasing availability and volume of financial information, the development of summarization algorithms that can provide short yet accurate information is of significant practical interest. To this end, the Financial Narrative Summarization (FNS)[1] challenge (Zmandar et al., 2022) intends to raise the quality of automated text summarization methods for the financial domain, for the Greek, English and Spanish languages. One of the main challenges for this task was the average length of the given annual reports (several dozens of pages), which made the training process extremely time consuming. In addition, the texts were extracted from PDF files with tables, charts, and numerical data, which resulted in poor, noisy inputs.

2. Past Work

The participating systems of previous editions of the challenge used techniques and methods ranging from rule-based extraction methods to high-performing deep learning models and word embeddings, including fine tuning pre-trained transformers models. Some teams investigated the hierarchy of the reports to select the narrative sections and identify the parts where the gold standard summaries were extracted. Participants applied techniques such as the Determinant Point Processes sampling algorithm (Kulesza and Taskar, 2012) or a combination of Pointer Network (Vinyals et al., 2015) and T5 (Raffel et al., 2019) algorithms. Others used word embeddings such as BERT embeddings (Devlin et al., 2018), word2vec, CBOW and skip grams ((Mikolov et al., 2013b), (Mikolov et al., 2013a)).

[1]The FNS challenge is part of the 4th Financial Narrative Processing Workshop

The best method in the previous edition (Orzhenovskii, 2021) was based on T5 (Raffel et al., 2019). The model was fine-tuned to generate the beginning of an abstractive summary and find the closest match of the output in the report's full text. The author also found intelligent insights in the data which simplified the problem, and much of our data treatment was based on those ideas.

3. Methodology

In this section we describe the different methodologies for each of the proposed languages. First, a preliminary analysis regarding the summaries with respect to the original reports they come from is presented, together with some considerations from the data analysis and exploration. Then, summarization models are explained for all three languages.

3.1. Previous Analysis and Considerations

We begin our analysis with the reports in English. In this case, as the summaries were extractive, a proper analysis was performed to detect where they began. For each report, a sentence tokenization was implemented using nltk's (Bird et al., 2009) *sent_tokenize* module. After the tokenization, the summaries were compared to the gold standard and the position where the gold included the sentence was saved. A few gold standards were given in the task but only the first one was used following the results on last year's competition (Orzhenovskii, 2021). In Figure 1 we can observe that very few reports start their summary after the 150-200th sentence. We performed this analysis based on the insights obtained by (Orzhenovskii, 2021). This can be used to optimize further processes as summaries usually start before the 250th sentence. The mean of the beginnings was between sentence 39 and 40.

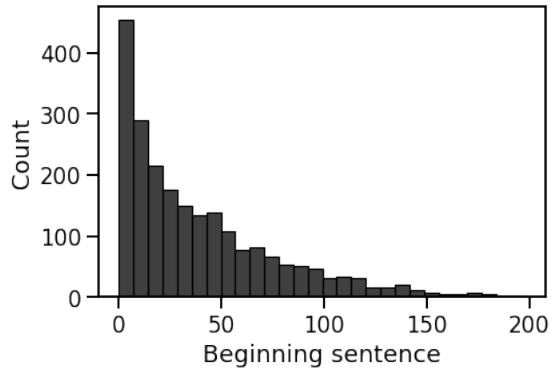

Figure 1: Histogram of beginnings of summaries for the English task.

As Spanish and Greek summaries were abstractive, no further analysis and considerations were taken into account.

3.2. Models

In this section we introduce the summarization models we used for this task, separately for each language.

3.2.1. English

For the English language, financial summaries are mostly CEO letters explaining the general results of the company as stated in the financial report. This means that summaries are literally contained in the original text, therefore the solution to this task could be extractive. This greatly simplifies the problem of generating the summaries, as no abstractive generative model is needed. The task is therefore reduced to finding where the summary (the CEO letter) starts and ends. Before this, a simpler approach was tried, based on classifying whether each sentence is part of the summary or not. This, however, proved to be too simplistic, therefore the alternative strategy was used.

There are various approaches to finding the start and end of the summary in the original text. One possible approach is to frame the summarization problem as a token classification task, where all tokens are null except for summary start and end tokens. This, however, poses a difficult learning problem. The learning signal becomes too sparse, since only one start and one end token are present in each document.

In this work we propose solving this task as a sequence classification problem, where the objective is to find the sentence where the summary starts. Given the distribution of real summaries in the train set, where it was observed that many of them were longer than 1000 words, and the workshop restriction of 1000 words per summary, the end of summaries was heuristically selected by taking the next 1000 words after the start of the beginning sentence predicted by our model.

The following procedure was used to build the training dataset for our model. The objective was to provide the model not only with the sentence to analyze, but also with the surrounding ones, in order to give the model more context to decide whether that sentence is the start of the summary or not. To this end, we picked surrounding sentences (both preceding and following the sentence being processed) until the token limit of our model (512 tokens) is reached.

A special `[SEP]` token is added to mark the boundaries of the sentence that should be classified by the model, thus producing a text in the form "Sentences previous to the query. All sentences we can fit. `[SEP]` Sentence being processed `[SEP]` Sentences following the sentence to analyze. Can also be more than one; All sentences we can fit". This way, the model can contextualize the sentence being processed at the moment.

The model used for this task was Deberta-V3-large (He et al., 2021), as it performs significantly better than the rest of the Encoder-based large models (the ones most suitable for a classification task like this one). In (He et al., 2021), a comparative table for GLUE tasks against BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNET (Yang et al., 2019), ELECTRA (Clark et al., 2020) and DeBERTa (He et al., 2020) is presented, showing that it is the best performing model in 7 out of 8 tasks.

To easen the classification task, some heuristics based on the preliminary analysis of the data were also applied. As it was identified that summaries start mostly between sentences 7 and 200, only the first 250 sentences from each financial report were considered, both on the training and testing sets. This greatly accelerated training time and reduced the time needed for generating predictions. This was especially relevant, given the size of the original financial reports. Moreover, this avoided predictions of starting positions beyond the 250th sentence and therefore unlikely according to the distribution observed in the training and validation splits.

Regarding the hyperparameters used to train the model, we performed preliminary experiments using the hyperparameter spaces from (He et al., 2021), and then launched the final run with the best configuration found.

3.2.2. Spanish

In the case of Spanish, summaries were not extractive, and that made the task much harder than in English. Original texts were of similar length as English ones, but in this case no classification model could be used. As, given the existing technology, it was not possible to use whole financial reports to learn to generate whole CEO letters, a full transfer learning approach was followed. This procedure consisted of using the Spanish portion of a multilingual summaries dataset to train different models. Details about the training data and results will be specified in the Experiments section.

For abstractive summarization tasks, an Encoder-Decoder architecture such as BART (Lewis et al., 2019), Pegasus (Zhang et al., 2019), Prophetnet (Yan

Hyperparameter	Values
Learning Rate	(3e-5, 7e-5, log)
Num Train Epochs	7
Train Batch Size	{32, 64, 128}
Warmup Steps	{50, 100, 500, 1000}
Weight Decay	(0.0, 0.1)

Table 1: Hyperparameter space for abstractive summarization models in Spanish.

et al., 2020) or T5 (Raffel et al., 2019) is needed. However, there are no such models trained in Spanish, therefore other approaches were tried. On the one hand, two multilingual Encoder-Decoder models were trained. On the other hand, a new Encoder-Decoder model was created from an existing Encoder-only model.

As for the hyperparameters, Optuna (Akiba et al., 2019) was used for finding the best hyperparameter set. For each model, the hyperparameter space in table 1 was used for looking for the best setting.

3.2.2.1 MT5

MT5 (Xue et al., 2020) is a multilingual variant of T5 (Raffel et al., 2019) that was pre-trained on a new Common Crawl-based dataset covering 101 languages, on multiple tasks, including abstractive text summarization. We fine-tuned the MT5 model on the Spanish portion of the MLSUM dataset (Scialom et al., 2020), to predict the concatenation of the title and the summary of each item in the dataset. We made the fine-tuned model available[2] at the huggingface hub.

3.2.2.2 XLM-Prophetnet

XLM-Prophetnet (Yan et al., 2020) is a cross-lingual version of ProphetNet, pretrained on wiki100 xGLUE dataset (Liang et al., 2020). Prophetnet is an Encoder-Decoder architecture suitable for sequence-to-sequence tasks. In English, it is able to perform similarly to BART (Lewis et al., 2019), T5 (Raffel et al., 2019), or Pegasus (Zhang et al., 2019) on abstractive summarization tasks, therefore its multilingual version is expected to work decently for the task proposed in this work. In this work, a fine-tuned version on the Spanish portion of MLSUM dataset[3] was made publicly available.

3.2.2.3 MariMari

(Rothe et al., 2019) propose to use already trained only-Encoder language models to create new Encoder-Decoder architectures from them. Their hypothesis is that much of the knowledge of such models could be reused for NLG tasks, given their great language modeling results and their good performance in NLU tasks. For that, two Encoder models are used, one as the Encoder and the other as the Decoder, including some cross-attention weights from one to the other.

Although there are no high-performing, openly available Encoder-Decoder models in Spanish, there are several Encoder-only models. After studying the different alternatives, which were compared in (Gutiérrez-Fandiño et al., 2021), we decided to use the Roberta-base from (Gutiérrez-Fandiño et al., 2021), also known as *MarIA*. Since our model is made up of two *MarIA* models, we decided to name it in a befitting way as *MariMari*. Moreover, Encoder-Decoder versions of BETO (Cañete et al., 2020), a Spanish BERT, had already been published, therefore we had a model to compare our own results against.

In (Rothe et al., 2019) the authors tested different configurations for their Encoder-Decoder models. Authors report the best configuration is to tie weights of the Encoder and the Decoder, which also has the advantage of saving GPU memory; therefore we followed this recommended configuration when training MariMari. We also made this model [4] openly available in the Huggingface Hub.

3.2.3. Greek

For the Greek language, the challenge was the lack of models available and the short time to train a big, state-of-the-art Greek language model. Also the debugging of the models posed an additional challenge, as no member of the team was a Greek speaker.

In order to tackle this, our approach consisted of a translation-summary-translation system that uses an existing Greek-English translation novel model (Tiedemann and Thottingal, 2020) based on MarianMT framework (Junczys-Dowmunt et al., 2018) and an English BART (Lewis et al., 2019) model which is particularly effective on summarization, translation and text generation in general. The checkpoint of the BART model used was fine-tuned on CNN Daily Mail, a large collection of text-summary pairs which suits our need on this specific task.

The last step on the task is the translation back to Greek. For this task, the DeepL API (DeepL, 2022) was used as the transformers-based solution by (Tiedemann and Thottingal, 2020) generated poor quality outputs such as continuously repeated or non-existing words.

4. Experiments and Results.

This section focuses mainly on the systems for English and Spanish, as these were the two languages for which experiments were carried out. For Greek, as explained in previous section, we decided to use already available methods without further training.

[2] https://huggingface.co/IIC/mt5-spanish-mlsum

[3] https://huggingface.co/IIC/xprophetnet-spanish-mlsum

[4] https://huggingface.co/IIC/marimari-r2r-mlsum

model	rouge1	rouge2	rougeL	rougeLsum
MT5	21.98	6.52	17.74	18.98
XML-Prophetnet	25.12	8.48	20.62	19.65
Mari-Mari	**28.78**	**10.67**	**23.04**	**25.78**
beto2beto	25.86	8.91	21.24	21.59

Table 2: Results on the test set of MLSUM for the MT5, XML-Prophetnet and Mari-Mari models presented in this work and the exisiting beto2beto model. Higher is better.

model	rouge1	rouge2	rougeL	rougeLsum
Mari-Mari	30.85	**10.36**	**14.92**	**29.35**
XLM-Prophetnet	**31.67**	10.10	14.74	27.51
MT5	30.38	9.12	14.31	28.03
beto2beto	31.50	9.97	14.56	27.69

Table 3: Results on the Spanish validation set of FNS for the MT5, XML-Prophetnet and Mari-Mari models presented in this paper and the exisiting beto2beto model. Higher is better.

4.1. Abstractive Summarization on MLSUM for Spanish

Our summarization models were trained on the Spanish portion of MLSUM (Scialom et al., 2020), since it is a large collection of text-summary pairs. We show the results of our models, and also of the model beto2beto-mlsum[5], on the test set of MLSUM, in Table 2.

We first report results on the test set of MLSUM, and then present results for the validation set of the FNS in Spanish.

We proceeded as follows. Once all three models were trained on MLSUM (Scialom et al., 2020), we split the reports into shorter segments that we could input in the models and produced summaries of each of the segments. If the concatenation of the resulting summaries was too long, we repeated the procedure with the summaries.

The summaries were also postprocessed, since the models had learnt certain sentences that were repeated throughout the MLSUM dataset.

Finally, we chose the Mari-Mari model, since the resulting summaries had higher scores on the validation set.

Table 3 shows the results of the three fine-tuned models on the Spanish validation set.

4.2. Binary Classification for Summary Start Detection in English.

The task for the English model is a binary classification task, of whether the current sentence starts or not the summary, therefore it is highly unbalanced, as only one sentence per report has label 1. For this reason, f1-macro (Opitz and Burst, 2019) is the metric selected to evaluate this model. Note that even when the model fails to detect the summary start correctly, if the start sentence predicted and the real one are close enough the resulting Rouge metric (Lin, 2004) on the summary will not be too penalized.

Table 4 shows results for Deberta-V3-large (He et al., 2021) on the validation set of FNS, in terms of F1-macro in detecting the start of the summaries.

[5]https://huggingface.co/LeoCordoba/beto2beto-mlsum

Metric	Deberta-v3-large
F1-Macro	**0.6989**

Table 4: F1-Macro for Deberta-v3-large on validation set of FNS.

	English	Spanish	Greek
ROUGE 2	36.6	12.5	9.5

Table 5: Results (ROUGE 2 F1 scores) on the test sets of our models, provided by the FNS organizers. Higher is better.

4.3. Results on the test sets

Table 5 shows the results (ROUGE 2 F1 scores) of our three models on the test sets (provided by the FNS organizers).

5. Conclusions

In this work we present several solutions for the FNS task of FNP 2022. First, extractive summarization models were trained in English. For that, most relevant Encoder-only language models in English were reviewed, selecting Deberta-v3-large in the end due to its effectiveness in English benchmarks.

A different approach was followed for Spanish and Greek. For Spanish, three different abstractive summarization models were trained, and their results are reported, both on the test set of MLSUM and the validation set of FNS. They are also compared against beto2beto, an existing model of similar size and architecture as the ones presented. Finally, for Greek, pretrained summarization models in English were used, together with automatic translation.

6. Acknowledgements

We would like to acknowledge the Cátedra of Computational Linguistics from Universidad Autónoma de Madrid [6] for letting us use their machines to train the extractive and abstractive summarization models.

7. Bibliographical References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902.

[6]http://catedras.iic.uam.es/catedra-linguistica-computacional/

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., and Pérez, J. (2020). Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Clark, K., Luong, M., Le, Q. V., and Manning, C. D. (2020). ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

DeepL. (2022). Deepl api. `https://www.deepl.com/es/docs-api`. Accessed: 2022-04-13.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., Gonzalez-Agirre, A., Armentano-Oller, C., Penagos, C. R., and Villegas, M. (2021). Spanish language models. *CoRR*, abs/2107.07253.

He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

He, P., Gao, J., and Chen, W. (2021). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July. Association for Computational Linguistics.

Kulesza, A. and Taskar, B. (2012). *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., Hanover, MA, USA.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Liang, Y., Duan, N., Gong, Y., Wu, N., Guo, F., Qi, W., Gong, M., Shou, L., Jiang, D., Cao, G., Fan, X., Zhang, B., Agrawal, R., Cui, E., Wei, S., Bharti, T., Qiao, Y., Chen, J., Wu, W., Liu, S., Yang, F., Majumder, R., and Zhou, M. (2020). XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *CoRR*, abs/2004.01401.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Opitz, J. and Burst, S. (2019). Macro F1 and macro F1. *CoRR*, abs/1911.03347.

Orzhenovskii, M. (2021). T5-LONG-EXTRACT at FNS-2021 shared task. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 67–69, Lancaster, United Kingdom, 15-16 September. Association for Computational Linguistics.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Rothe, S., Narayan, S., and Severyn, A. (2019). Leveraging pre-trained checkpoints for sequence generation tasks. *CoRR*, abs/1907.12461.

Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., and Staiano, J. (2020). Mlsum: The multilingual summarization corpus. *arXiv preprint arXiv:2004.14900*.

Tiedemann, J. and Thottingal, S. (2020). OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal, November. European Association for Machine Translation.

Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer networks. In C. Cortes, et al., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

Yan, Y., Qi, W., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., and Zhou, M. (2020). Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *CoRR*, abs/2001.04063.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Zmandar, N., El-Haj, M., Rayson, P., Abura'Ed, A., Litvak, M., Giannakopoulos, G., Pittaras, N., Carbajo-Coronado, B., and Moreno-Sandoval, A. (2022). The financial narrative summarisation shared task (fns 2022). In *Proceedings of the 4th Financial Narrative Processing Workshop*, Marseille, France, 24June. The 13th Language Resources and Evaluation Conference, LREC 2022.