POLITECNICO DI TORINO
Repository ISTITUZIONALE

End-to-end Training For Financial Report Summarization

Original
End-to-end Training For Financial Report Summarization / La Quatra, Moreno; Cagliero, Luca. - ELETTRONICO. - Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation:(2020), pp. 118-123. (Intervento presentato al convegno FNP @ COLING 2020 tenutosi a Barcelona, Spain (Online) nel 8/12/2020 - 13/12/2020).

Availability:
This version is available at: 11583/2860300 since: 2021-01-11T14:14:15Z

Publisher:
COLING

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

24 April 2024

End-to-end Training For Financial Report Summarization

Moreno La Quatra
Politecnico di Torino
moreno.laquatra@polito.it

Luca Cagliero
Politecnico di Torino
luca.cagliero@polito.it

Abstract

Quoted companies are requested to periodically publish financial reports in textual form. The annual financial reports typically include detailed financial and business information, thus giving relevant insights into company outlooks. However, a manual exploration of these financial reports could be very time consuming since most of the available information can be deemed as non-informative or redundant by expert readers. Hence, an increasing research interest has been devoted to automatically extracting domain-specific summaries, which include only the most relevant information.

This paper describes the *SumTO* system architecture, which addresses the Shared Task of the Financial Narrative Summarisation (FNS) 2020 contest. The main task objective is to automatically extract the most informative, domain-specific textual content from financial, English-written documents. The aim is to create a summary of each company report covering all the business-relevant key points.

To address the above-mentioned goal, we propose an end-to-end training method relying on Deep NLP techniques. The idea behind the system is to exploit the syntactic overlap between input sentences and ground-truth summaries to fine-tune pre-trained BERT embedding models, thus making such models tailored to the specific context. The achieved results confirm the effectiveness of the proposed method, especially when the goal is to select relatively long text snippets.

1 Introduction

Analyzing the annual financial reports is the most established way to assess the health state of business companies. For example, rating agencies, banks, and hedge funds rely on the information extracted from domain-specific reports to assign ratings, grant loans, and drive investment strategies (Piotroski, 2000). Unfortunately, the content of the released financial reports is highly redundant as it typically includes contextual and technical information that is marginally relevant to domain experts. The Shared Task of the Financial Narrative Summarization (FNS) research challenge (El-Haj et al., 2020) aims to address this issue by fostering innovative research on the problem of automatic extraction of domain-specific summaries from the annual financial reports.

The algorithms designed for automatic text summarization can be partitioned into two main classes: *Extractive* approaches and *Abstractive* ones. While extractive approaches pick existing text snippets (e.g., sentences, phrases, keywords) directly from the source text, abstractive methods generate new content based on the analysis of the input documents. The summarization process can be either *supervised*, when a portion of document content already annotated by human experts as relevant or not is available, or *unsupervised* when no a priori knowledge is given. The FNS shared task promotes the study, development, and testing of automated sentence-based summarization techniques tailored to the financial domain. To extract relevant sentences from annual financial reports, it provides researchers with a large set of humanly annotated data (El-Haj, 2019). Therefore, the present work addresses the study of a *supervised, extractive, sentence-based approach* to address the FNS Shared Task.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 118–123
Barcelona, Spain (Online), December 12, 2020.

Extractive summarization methods have found application in several domains, such as the summarization from news articles (e.g., (See et al., 2017; Cagliero et al., 2019; Krishnan et al., 2019)), scientific papers (e.g., (Cagliero and La Quatra, 2020; Cohan and Goharian, 2018; Collins et al., 2017)) and product reviews (i.e., (Ganesan et al., 2010)). Wide-ranging overviews of the state-of-the-art works on text summarization can be found in (Widyassari et al., 2020; Cagliero et al., 2020; El-Kassas et al., 2020). Using Machine Learning techniques to summarize documents entails (i) extracting relevant text features at the sentence level and (ii) feeding the extracted features to a supervised model to produce a sentence rank (El-Kassas et al., 2020). To address the former step, latent text representations based on Deep Learning models have shown to be very effective in generating relevant text features (Chen and Nguyen, 2019; Kobayashi et al., 2015) However, pre-trained deep NLP models need to be tailored to the specific context under analysis (e.g., medical data (Lee et al., 2020; Huang et al., 2019)), patent-related areas (Lee and Hsiang, 2019)). Previous works that use deep language models in the financial domain focused on the sentiment analysis task (Yang et al., 2020). To the best of our knowledge, this is the first attempt to fine-tune pre-trained deep NLP models in order to enhance the quality of the process of financial report summarization.

Section 2 overviews the architecture of the proposed summarizer. Section 3 and 4, 5 separately describe each phase of the summarization process. Section 6 summarizes the outcomes of the evaluation step. Finally, Section 7 draws conclusions and envisions future research steps.

2 The SumTO System

The *Sum*marizer based on end-*TO*-end training (SumTO) consists of a three-phase process, which is depicted in Figure 1. It comprises (i) a *preprocessing phase*, which transforms the raw textual documents and annotates the content at the sentence-level. (ii) a *training step*, which extract relevant concepts and relationships according to two established Deep language models, i.e., BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2019). (iii) a *evaluation step*, which rates the sentences of each test document according to the fine-tuned models trained at the previous step and produce a per-document summary consisting of the highly rated sentences.

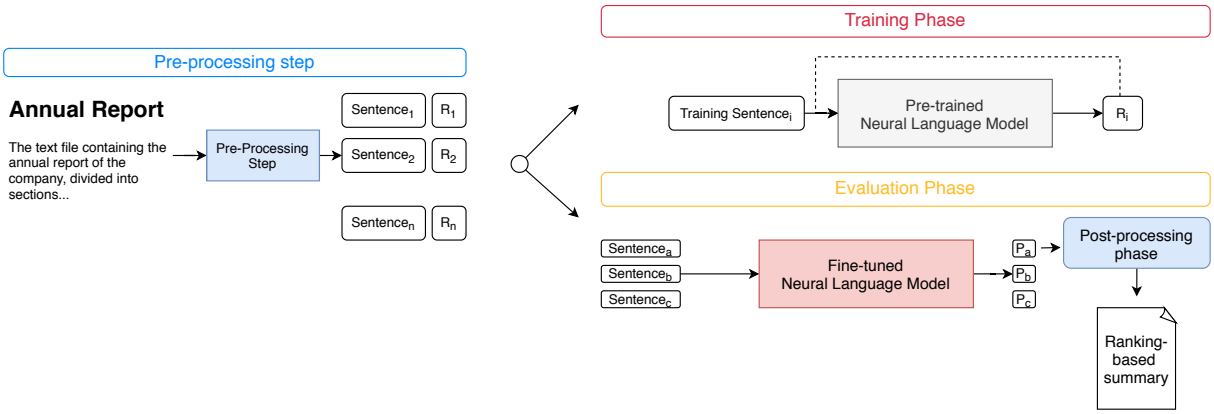

Figure 1: Outline of the proposed method

The fine-tuned model is able to provide better contextual representations for domain-specific vocabulary. The end-to-end process aims at training the model for the identification of relevant topics in the financial domain.

3 Data Collection and preprocessing

The data collection provided by the organizers of the FNS 2020 Shared Task includes the (i) the *training set*, consisting of 3,000 annual reports and 9,873 golden summaries (3.29 summaries per report, on average), (ii) the *evaluation set*, consisting of 363 annual reports and 1,250 golden summaries (3.44 summaries per report, on average), and (iii) the *test set*, consisting of 500 annual reports and 1,673 *blind*

System ID	Pre-Trained Model	Parameters Settings
1pe	`distilbert-base-cased`	N. of epochs: 1, Batch Size: 32, Learning rate: 2e-5
2pe	`distilbert-base-cased`	N. of epochs: 2, Batch Size: 32, Learning rate: 2e-5
3pe	`bert-base-cased`	N. of epochs: 1, Batch Size: 32, Learning rate: 2e-5

Table 1: System configuration settings

golden summaries (3.34 summaries per report, on average). The size of the training data enables the use deep Natural Language Processing models (Kobayashi et al., 2015).

The textual content of the reports in the training, evaluation, and test sets is transformed by applying the following data preparation steps.

1. *Text cleaning*: the source text, parsed from PDF documents, usually contains small errors in text parsing (e.g., a single word that spans over multiple lines is split in two different tokens). By employing ad-hoc regular expressions, the original content of each report is re-assembled as a single textual document.

2. *Sentence splitting:* The text stream is split into sentences by using *PunktSentenceTokenizer* provided by the Natural Language ToolKit (Loper and Bird, 2002) library.

3. *Data Annotation*: The sentence of the reports in the *training* and *evaluation* sets are annotated with the corresponding relevance score. The score indicates the similarity of the sentence with the content of the human-annotated summaries. It is computed by maximizing the syntactic overlap (i.e., Rouge-2 precision values (Lin, 2004)) with respect to all the given summaries[1].

4 Training phase of the Deep language model

A regression model is trained on the sentences of the training documents in order to predict the previously assigned sentence label (i.e., the Rouge-2 precision score). This idea behind to optimize the sentence relevance score according to the provided human annotation by fine-tuning the pre-trained BERT model (Devlin et al., 2019).

The overall architecture is trained using the Mean-Square loss and the *AdamW* optimizer (Loshchilov and Hutter, 2017) for faster convergence. Table 1 reports the settings for each system run. We generated three different fine-tuned models, hereafter denoted as *1pe*, *2pe*, *3pe*. The best performing model (i.e. *3pe*) and the code to apply the summarization algorithm are available on GitHub[2].

5 Evaluation phase

For each test document, the summarizer evaluates and ranks the corresponding sentences according to the fine-tuned model. Specifically, the sentences of the input report are forward-passed through the trained model and sorted in order of decreasing predicted Rouge-2 Precision score. The ranked list is post-processed by removing (i) duplicate sentences, (ii) sentences containing more than 50% of uppercase characters, (iii) sentence containing more than 50% of non alphabetic characters, (iv) sentences shorter than 5 words. The text snippets are selected from the post-processed pool according to their assigned score until the summary length requirement (up to 1000 words) is met. The output summary is generated by concatenating the post-processed sentences in order of decreasing relevance score.

6 Results

The output summaries submitted to the FNS 2020 Shared Task contest were evaluated by the shared task organizers. To evaluate the system outputs provided by the participants, they exploited the JRouge package[3], which is a lightweight, multilingual tool implementing the Rouge metrics (Lin, 2004).

[1] Each report may be annotated by multiple summaries provided by different experts.
[2] `https://github.com/MorenoLaQuatra/SumTO_financial_summarization`
[3] `https://bitbucket.org/nocgod/jrouge/wiki/Home`

Evaluation metric	Shared Task Results (3pe)	F1 (3pe)	F1 (2pe)	F1 (1pe)
Rouge 1	7th out of 14	**0.424**	0.422	0.421
Rouge 2	5th out of 14	**0.249**	0.235	0.237
Rouge SU4	5th out of 14	**0.264**	0.252	0.254
Rouge-L	3rd out of 14	**0.394**	0.385	0.387

Table 2: Systems results for the FNS 2020 Shared Task. The results of the best performing system are highlighted in bold.

Summaries were evaluated using the Rouge-1, Rouge-2, Rouge-SU4, and Rouge-L metrics. Beyond the systems proposed by the contest participants, the following baseline methods have been considered: (i) TextRank (Mihalcea and Tarau, 2004), (ii) LexRank (Erkan and Radev, 2004), (iii) POLY (Litvak and Vanetik, 2013), and (iv) a topline algorithm, i.e., MUSE (Litvak et al., 2010). Table 2 summarizes the F1-Score results achieved by our submitted runs. The scores of the best performing model (*3pe*) are reported in bold.

The SumTO system achieved fairly good results in terms of Rouge-L (i.e., finding the longest N-gram match with the ground truth), because our system tends to prefer relatively longer sentences. For the same reason, ROUGE-1 performance is on average worse than that achieved for Rouge-2 and rouge-SU4.

6.1 Computational requirements and execution times

The models were trained on a machine equipped with Intel® Xeon® Gold 5115 CPU, NVIDIA® Tesla® V100 16GB GPU and 512GB of RAM. Using this configuration the fine-tuning of the BERT model (on the full training set) took on average 36 hours per epoch, wheras for DistilBERT each epoch took less than 20 hours. During the evaluation phase, the summarization of a single annual report took around 30 seconds.

7 Conclusions and future research steps

The paper described an extractive summarization approach to summarizing textual financial reports. The proposed approach relies on the fine-tuning of a BERT deep language model. The goal is to deeply tailor the Deep NLP model to the specific context under analysis. The system runs were submitted to the FNS 2020 Shared Task, achieving fairly high performance in terms of Rouge-L score.

Our future research agenda will cover the following aspects:

Pruning of redundant information: The current summarization architecture is not able to prune redundant content, with respect to the previously selected sentences, during the sentence evaluation phase. We plan to extend system by embedding ad hoc redundancy penalty score.

Deeper model contextualization: The results have confirmed the effectiveness of the BERT architecture to support text summarization. We aim to explore the applicability of larger and deeper neural language models in order to better capture the semantic meaning of the analyzed sentences.

Acknowledgements

The research leading to these results is supported by the SmartData@PoliTO center for Big Data technologies.

References

Luca Cagliero and Moreno La Quatra. 2020. Extracting highlights of scientific articles: A supervised summarization approach. *Expert Systems with Applications*, 160:113659.

Luca Cagliero, Paolo Garza, and Elena Baralis. 2019. Elsa: A multilingual document summarization algorithm based on frequent itemsets and latent semantic analysis. *ACM Trans. Inf. Syst.*, 37(2), January.

Luca Cagliero, Paolo Garza, and Moreno La Quatra. 2020. Combining machine learning and natural language processing for language-specific, multi-lingual, and cross-lingual text summarization: A wide-ranging overview. In *Trends and Applications of Text Summarization Techniques*, pages 1–31. IGI Global.

L. Chen and M. L. Nguyen. 2019. Sentence selective neural extractive summarization with reinforcement learning. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–5.

Arman Cohan and Nazli Goharian. 2018. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2-3):287–303.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada, August. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mahmoud El-Haj, Ahmed AbuRa'ed, Nikiforos Pittaras, and George Giannakopoulos. 2020. The Financial Narrative Summarisation Shared Task (FNS 2020). In *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020*, Barcelona, Spain.

Mahmoud El-Haj. 2019. Multiling 2019: Financial narrative summarisation. In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, pages 6–10.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2020. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, page 113679.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. Association for Computational Linguistics.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization based on embedding distributions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1984–1989, Lisbon, Portugal, September. Association for Computational Linguistics.

D. Krishnan, P. Bharathy, Anagha, and M. Venugopalan. 2019. A supervised approach for extractive text summarization using minimal robust features. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 521–527.

Jieh-Sheng Lee and Jieh Hsiang. 2019. Patentbert: Patent classification with fine-tuning a pre-trained bert model. *arXiv preprint arXiv:1906.02124*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Marina Litvak and Natalia Vanetik. 2013. Mining the gaps: Towards polynomial summarization. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 655–660.

Marina Litvak, Mark Last, and Menahem Friedman. 2010. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 927–936.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Joseph D. Piotroski. 2000. Value investing: The use of historical financial statement information to separate winners from losers. *Journal of Accounting Research*, 38:1–41.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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

Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, and De Rosal Ignatius Moses Setiadi. 2020. Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*.

Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications.