Summarization of financial documents with TF-IDF weighting of multi-word terms

Sophie Krimberg
Shamoon College of
Engineering (SCE)
Beer-Sheva
Israel
krsofi@gmail.com

Natalia Vanetik
Shamoon College of
Engineering (SCE)
Beer-Sheva
Israel
natalyav@sce.ac.il

Marina Litvak
Shamoon College of
Engineering (SCE)
Beer-Sheva
Israel
marinal@ac.sce.ac.il

Abstract

Financial documents, such as corporate annual reports, are usually very long and may consist of more than 100 pages. Every report is divided into thematic sections or statements that have an inner structure and include special financial terms and numbers. This paper describes an approach for summarizing financial documents based on a Bag-of-Words (BOW) document representation. The suggested solution first calculates the Term Frequency-Inverse Document Frequency (TF-IDF) weights for all single-word and multi-word expressions in the corpus, then finds the sequence of words with a maximum total weight in each document. The solution is designed to meet the requirements of the Financial Narrative Summarization (FNS 2021) shared task and has been tested on FNS 2021 dataset shared-task dataset.

1 Introduction

Corporate annual reports and financial statements are challenging to summarize due to their length, format, structure, and contents. An annual report is a document of tens and often hundreds of pages. Sometimes annual report includes a table of contents, but there are a lot of reports that do not. Usually, reports have several thematic sections, but the order, the quantity, and the structure of sections differ from one report to another. Financial documents use specialized financial terms. Additionally, every company that publishes a report operates within its field, and this field's lexicon can appear in the report and be an important part of it, while all of the other documents in the corpus do not use that lexicon at all.

The 1st Joint Workshop on financial Narrative Processing and MultiLing financial Summarisation (FNP-FNS 2020) (El-Haj et al., 2020a) ran the financial narrative summarisation (FNS) task, which resulted in the first large-scale experimental results and state-of-the-art summarization methods applied to financial data. The task focused on annual reports produced by UK firms listed on the London Stock Exchange. Because companies usually produce glossy brochures with a much looser structure, this makes automatic summarization of such reports a challenging task. A total number of 9 teams participated in the FNS 2020 shared task with a total of 24 system submissions. All teams were ranked by several ROUGE-based measures and compared to the four topline and baseline summarizers—MUSE (Litvak et al., 2010), POLY (Litvak and Vanetik, 2013), TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004)—in (El-Haj et al., 2020b).

The participating systems used a variety of techniques and methods ranging from rule based extraction methods (Litvak et al., 2020; Vhatkar et al., 2020; Arora and Radhakrishnan, 2020; Azzi and Kang, 2020) to traditional machine learning methods (Suarez et al., 2020; Vhatkar et al., 2020; Arora and Radhakrishnan, 2020) and high performing deep learning models (Agarwal et al., 2020; Singh, 2020; La Quatra and Cagliero, 2020; Vhatkar et al., 2020; Arora and Radhakrishnan, 2020; Azzi and Kang, 2020; Zheng et al., 2020). The text representation was also very diverse among the participating systems—very basic morphological and structure features (Li et al., 2020; Suarez et al., 2020), syntactic features (Vhatkar et al., 2020), and semantic vectors using word embeddings (Agarwal et al., 2020; Suarez et al., 2020) were applied. In addition, some teams (Litvak et al., 2020; Zheng et al., 2020) investigated the hierarchical structure of reports. Different ranking techniques, such as Determinantal Point Processes (Li et al., 2020), a combination of Pointer Network and Text-to-text transfer Transformer algorithms (Singh, 2020) were used for extractive approaches, together with deep language models (La Quatra and Cagliero, 2020; Zheng et al., 2020), hierarchical summarization under different discourse topics (Litvak et al., 2020), and ensemble based models (Arora and Radhakrishnan, 2020) have also been reported. The main challenge of this task, as reported by its participants, was the average length of a document, which made the training process extremely inefficient. In addition, participants argued that extracting text and then structure from PDF files with numerous tables, charts, and numerical data resulted in a lot of noise.

This year FNS-2021 (El-Haj et al., 2021) shared task asks to provide summaries of annual company reports. The dataset is supplied with 2-4 gold standard summaries per document. These gold standard summaries are complete sections of the original document selected by human financial experts as the most important sections of the documents.

Term Frequency-Inverse Document Frequency (TF-IDF) (Sammut and Webb, 2010) is a term weighting scheme, commonly used for making relevant decisions and discover the strength of the relationship of words with the document they appear in (Ramos et al., 2003).

In this paper, we propose a TF-IDF weighing method that helps to determine the most successful candidate for the extractive summary among the possible continuous document parts of the required length. This approach is based on the fact that all of the gold standard summaries in the data provided by the organizers are in fact sections of the original documents that did not undergo any rewriting. We use the TF-IDF score to detect the most important sequence of up to 1000 words in a document. While the classic implementation is based on the evaluation of single words, we calculate the TF-IDF values for single-word and multi-word terms, mainly to recognize the specific financial terminology.

2 The method

On purpose to find sequences of up to 1000 most important words in every document of the corpus, we do the next steps:

1. define the value of the maximal length of multi-word term (See Section 2.1);

2. find all the existing multi-word terms in the corpus and calculate the TF-IDF score for every one of them;

3. compute summarized TF-IDF scores for all continuous sequences with 1000 words in a document;

4. select the highest-ranking sequence as a summary for the specific document.

The pipeline of our approach is depicted in Figure 1.

2.1 Multi-word terms

Because classic TF-IDF is computed for single-word terms only, and we want to extend it to multi-word terms, we introduce a parameter that defines a maximal number of words in such a term. The aim of evaluating multi-word terms is to recognize the set of important document-specific phrases from their TF-IDF weights.

2.2 Preprocessing

The original files are preprocessed using Python *nltk* library (Bird et al., 2009). The preprocessing includes text splitting, tokenization, special symbols removal, removing of phone numbers, emails etc. Stopwords are not removed, but we use a custom stopword list containing the words ['and', 'the', 'is', 'are', ' this','at', 'of', 'to', 'in', 'on', 'for', 'or','a', 'an', 'as', 'page', 'by', 'with', 'our', 'we', 'that', 'may']. All multi-word terms that contain stopwords only get zero TF-IDF values.

2.3 Creating the TF-IDF matrix of a document

When the maximal number of words in term is defined (denoted by TL), the system finds in the corpus all the existing word sequences of length 1 to TL and calculates the TF-IDF score for every one of them. The following steps are performed:

1. Generate multi-word terms of length 1 to TL as follows. For a document with DL words, there are DL single-word terms, $DL - 1$ two-word terms, and so on. Finally, we have $DL - TL + 1$ terms with TL words.

2. Let T be a multi-word term with $TL := |T|$ words in a document D_i having $DL_i := |D_i|$ words in total, and let T appear DR_i times in the document D_i. Term frequency of T in D_i is calculated as

$$TF(T, D_i) = \frac{DR_i}{DL_i - TL + 1} \quad (1)$$

3. Let T appear in CR documents in the corpus of size N. Then the IDF score of T is:

$$IDF(T) = \log \frac{N}{CR} \quad (2)$$

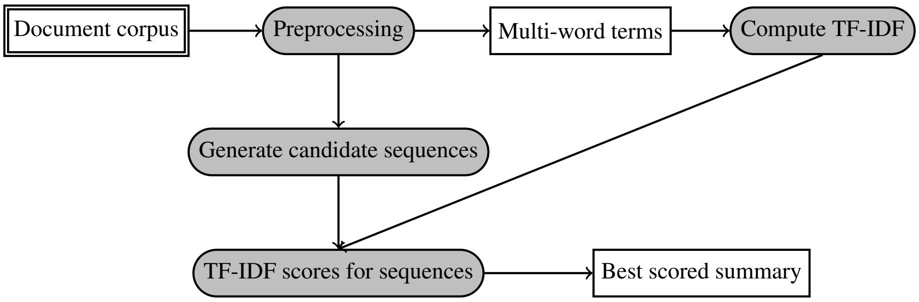

Figure 1: Pipeline of our approach

4. Finally, the TF-IDF score of a term T in document D_i is:

$$\textit{TF-IDF}(T, D_i) = \textit{TF}(T, D_i) \cdot \textit{IDF}(T) \quad (3)$$

2.4 Most important sequence in a document

In every document D_i, we find all the sequences of up to 1000 words (there are $DL_i - 999$ such sequences in a document with more than a 1000 words), and calculate the sum of TF-IDF values for all the multi-word terms of any length that appear in every such sequence S:

$$\textit{TF-IDF}(S) = \sum_{k=1}^{TL} \sum_{T \in S, |T|=k} \textit{TF-IDF}(T, D_i) \quad (4)$$

We rank the sequences by their TF-IDF scores and select the highest-ranking sequence as our summary. Implementation of calculating the totals for multiple sequences is based on the idea that given total of sequence $W_1 W_2 \ldots W_n$ we can calculate the total of sequence $W_2 W_3 \ldots W_{n+1}$ by subtracting the values of terms that can include W_1 and adding the values of terms that can include W_{n+1} (according to the maximum number of words in a term). This approach allows the system to calculate and compare thousands of such sequences in each document in a very short time.

3 Experiments

FNS 2021 Shared Task provides a dataset that contains companies' annual reports and 3-4 gold standard summaries for each report. The gold standard summaries were created by extracting whole sections (one or more) from the original document, according to a human financial expert's decision. The selected summaries sections are considered by the experts as most important and informative. Table 1 describes the dataset contents. The training dataset, which contains 3,000 reports and 9,873 gold summaries, was randomly divided by us into 3 groups of 1,000 documents each to facilitate the tf-idf computation. Furthermore, every one of those three groups was divided into two subgroups of 500 documents each. We three variants of our system using values 1, 2, and 3 as the multi-word term size TL.

3.1 Tools and runtime environment

For preprocessing such as sentence splitting and tokenization we used *nltk* package (Bird et al., 2009); We have used the MUSEEC tool (Litvak et al., 2016) to compute MUSE summaries to be used as a baseline a with 1000-word limits, respectively. We used the ROUGE 2.05 java package (Ganesan, 2018).

3.2 Methods and baselines

For evaluation of the results of this approach, we applied it on the validation part of the FNS 2021 shared task dataset and compared the results to the results of Muse (Litvak et al., 2016) on the same set of documents. As an additional reference, we use the results of a trivial TOP-K baseline that includes the first 1000 words of a document. The results are reported in table 2, the results of our approach appear as **TFIDF-SUM-N**, where the number N is the maximal number of words in a term. [1] Experiments were performed on Google Colab with the default configuration.

3.3 Evaluation results

Four ROUGE (Lin, 2004) metrics—ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-SU4 were applied on the validation set. Table 2 shows the results, with recall, precision, and F-measure for each metric. It can be seen that as the maximum number of words in the term increases, the results

[1] The results on the test set, provided by the FNS organizers, can be seen in the Appendix and on the leaderboard: https://www.lancaster.ac.uk/staff/elhaj/docs/fns2021_results.pdf.

dataset	# documents	# gold summaries	avg sentences	avg words	avg characters
Train	3,000	9,873	2,700	58,838	291,014
Validation	363	1,250	3,786	82,906	416,040
Test	500	NA	3,743	82,676	412,974

Table 1: FNS 2021 dataset statistics.

System	R1 R	R1 P	R1 F	R2 R	R2 P	R2 F
TOP-K	0.266	0.241	0.221	0.040	0.038	0.034
MUSE	0.261	0.297	0.243	0.042	0.052	0.040
TFIDF-SUM-1	0.353	0.317	0.322	0.153	0.110	0.121
TFIDF-SUM-2	0.450	0.396	0.410	0.244	0.156	0.183
TFIDF-SUM-3	**0.477**	**0.415**	**0.433**	**0.279**	**0.177**	**0.209**
System	**RL R**	**RL P**	**RL F**	**RSU4 R**	**RSU4 P**	**RSU4 F**
TOP-K	0.264	0.239	0.220	0.081	0.076	0.069
MUSE	0.255	0.292	0.238	0.084	0.100	0.079
TFIDF-SUM-1	0.263	0.279	0.258	0.218	0.141	0.164
TFIDF-SUM-2	0.374	0.332	0.343	0.312	0.188	0.227
TFIDF-SUM-3	**0.411**	**0.362**	**0.374**	**0.344**	**0.207**	**0.250**

Table 2: ROUGE results for FNS-2021 validation set.

System	R1 F	R2 F	RL F	RSU4 F
BASE	0.45	0.24	0.42	0.27
MUSE	0.50	0.38	0.52	0.43
TFIDF-SUM-1	0.33	0.12	0.27	0.17
LexRank	0.31	0.12	0.27	0.16

Table 3: ROUGE results for FNS-2021 test set.

improve, but even with a single term (TFIDF-SUM-1), the system outperforms the baselines. Due to time constraints, only the TFIDF-SUM-1 system was submitted to the FNS-2021 shared task competition and it appears in its results as an SCE-3 system.

It is important to note that increasing the maximum number of words in a multi-word term increases their amount drastically, and the memory usage increases as well. Therefore running the system with 3-word terms on Colab required us to divide the dataset into two parts and to compute the tf-idf scores for them separately. This approach reduces the precision of tf-idf, but because every run is still performed on almost 200 documents, we can see from the resulting ROUGE scores that an additional term compensates for the lack of tf-idf precision. Table 3 shows the results for the same ROUGE metrics, F-measure, obtained on the test set (provided by the FNS organizers).

3.4 Performance

Our system works very fast while producing hundreds of summaries in several minutes. For example, for 363 annual reports from Validation dataset, execution on Google Colab with default configuration was completed in 2 minutes 54 seconds with TFIDF-SUM-1, 6 minutes 22 seconds with TFIDF-SUM-2 and 10 minutes 50 seconds with TFIDF-SUM-3. Times may differ as the performance of Colab itself changes. But as the maximum number of words in a multi-word term increases, more possible terms exist and more memory is required. Using multi-word terms with more than three words resulted in an out-of-memory error.

4 Conclusions and Future Work

This paper introduces a method for summarization of financial documents. The method implements the TF-IDF technique with optimization for multi-word terms. The system is fast, simple, and outperforms baselines. The evaluation results show that (1) evaluating multi-word terms vs single-word ones improves the quality of the summaries and (2) that extracting continuous sequence from the document provides the results.

Future work may include modifying the current method to extract the most important sentences instead of extracting the whole sequence. In addition, combining the multi-term TF-IDF weighting scheme with machine learning algorithms and FinBERT (Yang et al., 2020) embedding may provide interesting results.

References

Raksha Agarwal, Ishaan Verma, and Niladri Chatterjee. 2020. Langresearchlab_nc at fincausal 2020, task 1: A knowledge induced neural net for causality detection. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 33–39.

Piyush Arora and Priya Radhakrishnan. 2020. Amex ai-labs: An investigative study on extractive summarization of financial documents. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 137–142.

Abderrahim Ait Azzi and Juyeon Kang. 2020. Extractive summarization system for annual reports. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 143–147.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Mahmoud El-Haj, Vasiliki Athanasakou, Sira Ferradans, Catherine Salzedo, Ans Elhag, Houda Bouamor, Marina Litvak, Paul Rayson, George Giannakopoulos, and Nikiforos Pittaras. 2020a. Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*.

Mahmoud El-Haj, Marina Litvak, Nikiforos Pittaras, George Giannakopoulos, et al. 2020b. The financial narrative summarisation shared task (fns 2020). In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 1–12.

Mahmoud El-Haj, Nadhem Zmandar, Paul Rayson, Ahmed AbuRa'ed, Marina Litvak, Nikiforos Pittaras, and George Giannakopoulos. 2021. The Financial Narrative Summarisation Shared Task (FNS 2021). In *The Third Financial Narrative Processing Workshop (FNP 2021)*, Lancaster, UK.

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. 2018. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*.

Moreno La Quatra and Luca Cagliero. 2020. End-to-end training for financial report summarization. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 118–123.

Lei Li, Yafei Jiang, and Yinan Liu. 2020. Extractive financial narrative summarisation based on dpps. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 100–104.

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

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.

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, Natalia Vanetik, Mark Last, and Elena Churkin. 2016. Museec: A multilingual text summarization tool. In *Proceedings of ACL-2016 System Demonstrations*, pages 73–78.

Marina Litvak, Natalia Vanetik, and Zvi Puchinsky. 2020. Sce-summary at the fns 2020 shared task. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 124–129.

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.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242(1), pages 29–48. Citeseer.

Claude Sammut and Geoffrey I. Webb, editors. 2010. *TF–IDF*, pages 986–987. Springer US, Boston, MA.

Abhishek Singh. 2020. Point-5: Pointer network and t-5 based financial narrative summarisation. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 105–111.

Jaime Baldeon Suarez, Paloma Martínez, and Jose Luis Martínez. 2020. Combining financial word embeddings and knowledge-based features for financial text summarization uc3m-mc system at fns-2020. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 112–117.

Amit Vhatkar, Pushpak Bhattacharyya, and Kavi Arya. 2020. Knowledge graph and deep neural network for extractive text summarization by utilizing triples. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 130–136.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *CoRR*, abs/2006.08097.

Siyan Zheng, Anneliese Lu, and Claire Cardie. 2020. Sumsum@ fns-2020 shared task. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 148–152.