Machine Learning with Applications 9 (2022) 100324

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Summarization of financial reports with TIBER

Natalia Vanetik, Marina Litvak *, Sophie Krimberg

Software Engineering Department, Shamoon College of Engineering, Bialik 56, Beer Sheva, Israel

ARTICLE INFO

Keywords:
Extractive summarization
Financial reports
Node embeddings
BERT

ABSTRACT

This paper reports an approach for summarizing financial texts that combine several techniques for sentence representation and neural document modeling. Our approach is extractive and it follows the classic pipeline of ranking and consequent selecting of the top-ranked text chunks. We evaluate our method on the financial reports provided in the Financial Narrative Summarization (FNS 2021) shared task. The data for the shared task was created and collected from publicly available UK annual reports published by firms listed on the London Stock Exchange. The reports composed FNS 2021 dataset are very long, have many sections, and are written in "financial" language using various special terms, numerical data, and tables. The results show that our approach outperforms the FNS topline with a very serious advantage. In addition to its performance, our approach is also time-efficient.

1. Introduction

All companies, big and small, usually produce a variety of reports containing both narrative and numerical information during their financial year, including annual financial reports. The general purpose of these reports is to provide information about the results of operations, financial position, and cash flows of an organization. This information is used by the readers of financial reports to make decisions regarding the allocation of resources (Bragg, 2021). However, it is quite difficult to keep track of vast amounts of financial information manually. Therefore, there is a vital need for automatic summarization systems to reduce the time and effort of both the shareholders and investors in decision making. A systematic literature review at Baviskar, Ahirrao, Potdar, and Kotecha (2021) reveals that AI-based approaches have a strong potential to extract useful information from unstructured documents automatically, which can save organizations millions of dollars every year.

This paper describes a new approach to the extractive summarization of financial reports. Our method utilizes a joint vector representation of sentences, based on n-grams, BERT sentence vectors, sentiment analysis, and neural node embedding computed from a sentence syntactic structure to rank consecutive chunks of source documents. The top-ranked chunks are then selected as summaries. Our contribution is three-fold: (1) we utilize different types—semantic, syntactic, structural, and sentiment—information to represent the input; (2) we use both prediction and heuristic weighting schemes to encode sentences' importance; (3) our method is computationally efficient which is especially important when dealing with long documents.

The structure of this paper is as follows. Section 2 describes the related work in the area of financial document summarization. Section 3 describes our approach—text representation (Section 3.4), prediction models (Section 3.5), summary generation algorithm (Section 3.6); methodological contribution is described in detail in Section 3.2. Section 4 describes the results of experimental evaluation, Section 5 describes the limitations of our study, and Section 6 summarizes with conclusions and future work.

2. Related work

There is a growing interest in the application of automatic and computer-aided approaches for extracting, summarizing, and analyzing both qualitative and quantitative financial data, as a series of FNP and related workshops (El-Haj, 2019; El-Haj, Litvak et al., 2020; El-Haj, Rayson, & Moore, 2018; Zmandar, El-Haj et al., 2021) recently demonstrates. However, before these workshops, only a few attempts were made to summarize financial reports (Isonuma, Fujino, Mori, Matsuo, & Sakata, 2017), while most works focused on the summarization of financial news (Baralis, Cagliero, & Cerquitelli, 2016; de Oliveira, Ahmad, & Gillam, 2002; Filippova, Surdeanu, Ciaramita, & Zaragoza, 2009; Yang & Wang, 2003; Zhang, Chen, & Xiao, 2018). It is needless to say that financial reports are very different from news articles in at least four parameters: length, structure, format, and lexicon.

The 1st Joint Workshop on financial Narrative Processing and MultiLing financial Summarization (FNP-FNS 2020) (El-Haj, Athanasakou et al., 2020) ran the financial narrative summarization (FNS) task, which resulted in the first large-scale experimental results and state-of-the-art summarization methods applied to financial data. The task has focused on annual reports produced by UK firms listed on the

* Corresponding author.
E-mail addresses: natalyav@ac.sce.ac.il (N. Vanetik), marinal@ac.sce.ac.il (M. Litvak), sofiak@ac.sce.ac.il (S. Krimberg).

https://doi.org/10.1016/j.mlwa.2022.100324
Received 3 December 2021; Received in revised form 4 May 2022; Accepted 4 May 2022
Available online 10 May 2022

London Stock Exchange (LSE). 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.

Rule-based methods, traditional machine learning techniques (such as MNB, SVM, etc.) and deep neural networks (such as BILSTM, CNN, GRUs, etc.), were adopted by the participating teams. In particular, rule based extraction methods were used in Arora and Radhakrishnan (2020), Azzi and Kang (2020), Litvak, Vanetik, and Puchinsky (2020) and Vhatkar, Bhattacharyya, and Arya (2020); traditional machine learning methods were adopted in Arora and Radhakrishnan (2020), Suarez, Martínez, and Martínez (2020) and Vhatkar et al. (2020); and high performing deep learning models were the focus of works (Agarwal, Verma, & Chatterjee, 2020; Arora & Radhakrishnan, 2020; Azzi & Kang, 2020; La Quatra & Cagliero, 2020; Singh, 2020; Vhatkar et al., 2020; Zheng, Lu, & Cardie, 2020).

The text representation was also very diverse among the participating systems. Basic morphological and structure features were applied in Li, Jiang, and Liu (2020) and Suarez et al. (2020), syntactic features were used in Vhatkar et al. (2020), semantic vectors using word embeddings were applied in Agarwal et al. (2020) and Suarez et al. (2020), and the hierarchical structure of reports was employed in Litvak et al. (2020) and Zheng et al. (2020).

The majority of the applied techniques were extractive and used a wide list of different ranking techniques. This is the obvious choice for long documents because global optimization techniques are much slower than ranking when the documents in question are very large. The ranking techniques used by competing systems include: Determinantal Point Processes sampling (Li et al., 2020), a combination of Pointer Network and T-5 (Test-to-text transfer Transformer) algorithms (Singh, 2020), deep NN language models (La Quatra & Cagliero, 2020; Zheng et al., 2020), importance ranking based on discourse topics (Litvak et al., 2020), and ensemble-based models (Arora & Radhakrishnan, 2020).

The next Financial Narrative Summarization Shared Task on summarizing UK annual reports was organized as part of the Financial Narrative Processing 2021 Workshop (FNP 2021 Workshop). The shared task included one main task which is the use of either abstractive or extractive automatic summarizers to summarize long documents in terms of UK financial annual reports. This shared task was the second to target financial documents. The data for the task was also created and collected from publicly available UK annual reports published by firms listed on the London Stock Exchange and it is an extension of the dataset released for the FNS 2020. A total number of 10 systems from 5 different teams participated in the FNS 2021 shared task. The participating systems used a variety of techniques and methods ranging from fine-tuning pre-trained transformers to using high-performing deep learning models and word embeddings. Some works investigated the hierarchy of the annual reports to extract the narrative sections and identify the parts in the report from which the gold summaries mostly were extracted. The majority of the applied techniques were extractive since the dataset is highly structured with discrete sections.

The variety of methods includes application of T-5 language model (Orzhenovskii, 2021), BERT-based representation (Gokhan, Smith, & Lee, 2021; Litvak & Vanetik, 2021), LSTM (Litvak & Vanetik, 2021), neural node embeddings (Litvak & Vanetik, 2021), unsupervised clustering (Gokhan et al., 2021) and extraction of word sequences (n-grams) with the maximal tf–idf (term frequency inverse document frequency) weights (Krimberg, Vanetik, & Litvak, 2021). An end-to-end hybrid extractive-abstractive training method using pointer network generators and reinforcement learning was applied in Zmandar, Singh, El-Haj, and Rayson (2021).

We have seen from Litvak and Vanetik (2021) that while using document-wise neural node embeddings is beneficial in terms of the scores, it is quite time-consuming. Therefore, we decided to use sentence-wise node embeddings in our work to improve the runtime.

The winning system, T5-LONG-EXTRACT (Orzhenovskii, 2021) based on pre-trained language model T5, was ranked 1st on Rouge-1, Rouge-2, Rouge-SU4, and Rouge-L metrics.

Two participating systems have managed to outperform the topline system, MUSE (Litvak, Last, & Friedman, 2010).

One of the main challenges and limitations reported by the participants of both competitions was the average length of annual reports (around 60,000 words), 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. These limitations open up an interesting research problem that is worth investigating.

Another observation is that focusing on a syntactic or semantic text representation alone results in information loss—therefore, in our work, we employ both types of representation to encode sentence meaning and its structure.

Most recent works outside the FNS shared tasks propose to summarize long financial reports with neural models. For example, in Agrawal, Anand, Arunachalam, and Varma (2021) and Agrawal, Anand, Gupta, Arunachalam, and Varma (2021) a Goal Guided Summarization (GGS) technique was proposed, where the goal is the decision to buy or sell a company's shares. The authors used hierarchical neural models for extracting summaries. The extrinsic evaluation showed that the summaries generated by the GGS model the decision of buying and selling shares better than summaries produced by other summarization techniques or the complete documents. Both intrinsic and extrinsic evaluations were performed using the 117,452 U.S. Securities and Exchange Commission (SEC) 10-K reports from 1994 to 2018 from 11,476 different companies. A dataset of multi-layout unstructured invoice documents was proposed and evaluated with a Bidirectional LSTM and its combination with Conditional Random Fields in Baviskar, Ahirrao, and Kotecha (2021). The work suggests that the dynamic word embeddings with the embedding layer of the Neural Network model perform better than pre-trained word embeddings, which do not provide the proper vector representation of unique words present in invoice documents. The insights of this work comply with our decision to generate our embeddings for report sentences and with our choice of a sentence classification model.

3. The method

In this section, we describe our method for financial summarization, called TIBER (standing for **T**f **I**df **B**ert d**E**pendency p**R**ediction), that utilizes multiple types of text representation for ranking the consequent sequences of sentences (chunks) and then selecting the top-ranked chunks as summaries.

3.1. The pipeline

Our method generates three representations of sentences—n-grams, BERT vectors with sentiment data (polarity and subjectivity), and neural node embeddings encoding the sentence dependency structure. We use the n-grams as dimensions in tf–idf vectors, and the other two representations as a setup for neural sentence classification models that produce sentence labels. The sentence labels are then used as additional dimensions in our joint vector representation of sentences.

We generate summary candidates by examining consecutive chunks of source documents and ranking them according to their weights produced by the three sentence representations. The length of chunks equals the length of the desired summary (for the FNS 2021 dataset, this length is set to 1,000 words.) The top-ranked document consecutive chunk then becomes our summary. The pipeline of our approach is depicted in Fig. 1.

Our motivation behind this joint representation was to consider as much information of different types—statistical, semantic, syntactic, and even sentiment—in the ranking procedure while keeping a reasonable runtime.

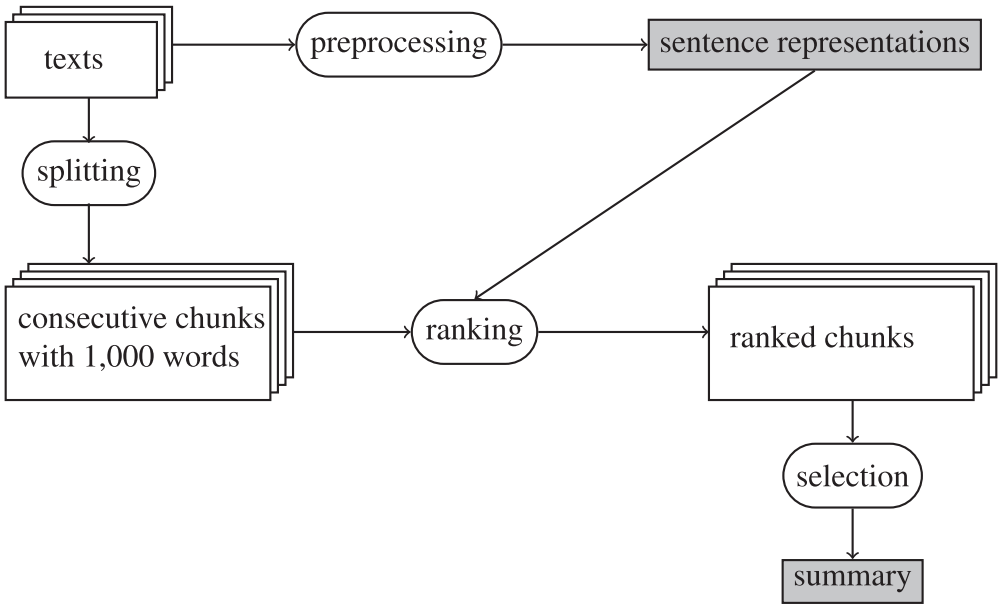

Fig. 1. The general pipeline of TIBER.

3.2. Methodological contribution

By considering n-grams instead of bag-of-words, we want to catch important multi-word terminology from the financial domain (for example: *"Unlevered Free Cash Flow"*, *"Applicable Federal Rate"*, *"Annual Percentage Rate"*, etc.). By including BERT vectors we consider the semantics of the examined text. Following the important observation about sentiment features in financial analysis (Li, Shi, Wang, & Zhou, 2021), we consider the sentiment polarity and subjectivity of each sentence. In addition, we consider the syntactic structure of the examined sentences to learn about sentences with typical grammatical structures in financial summaries.

However, not all types of information can be directly aggregated and transformed into a rank. For example, sentiment labels can have hints about a sentence's importance for a summary, but it needs to be learned from annotated data by a supervised model. The same can be said about semantic BERT vectors and syntactic encoding. Therefore, we apply these types of information as an input for classification models which classify sentences to summary (important) and other sentences, and then use labels they produce as direct importance indicators, along with tf–idf scores of n-grams contained in a text chunk.

To achieve the second goal of reasonable runtime, we separated classification models based on semantic and syntactic encoding. As our experiments approved (see below), both models have the same accuracy but different runtimes—computing node embeddings is an additional stage that is absent in a model utilizing semantic representation (we used pre-trained BERT vectors). In addition, we wanted to see whether we can prefer one of the prediction models and, as result, avoid redundant computations. Applying one classification model on joint vectors would deprive us of this choice and consume much more time and memory. Another dilemma was about sentiment labels—which model do they belong to? Our choice is intuitively motivated by the "nature" of sentiment labels, which are derived from semantics but cannot be directly obtained from syntax. Therefore, we added two sentiment labels—polarity and subjectivity—to BERT vectors.

3.3. Preprocessing

First, we only kept the first 10% of each file in the training and validation sets of the FNS-2021 shared task dataset (Zmandar, El-Haj et al., 2021) available at http://wp.lancs.ac.uk/cfie/fns2021/ following the insight of Gokhan et al. (2021). As Gokhan et al. (2021) have discovered, most of the gold summaries for the training set of FNS-2021 are contained in the first 10% of a document. Further examination has confirmed that this is indeed the case for all but the 22 documents out of 3,000 documents in the training set. Shortening of documents allows us to train our neural models (see Section 3.5 below) in a reasonable time. We eliminate empty sentences and very short (2 words or less) sentences but do not perform any additional data cleaning.

We perform sentence splitting, tokenization, and dependency parsing. We store the following information for every sentence in a truncated document: (1) BERT sentence vector for the entire sentence; (2) tokens; (3) word vectors of tokens; (4) a tree of syntactic dependencies for every sentence; (5) sentiment data for every sentence that includes sentence polarity and subjectivity as numeric values.

3.4. Sentence representation

Below, we describe three types of sentence representation utilized by our summarization approach.

3.4.1. Representation with n-grams

We operate n-grams of words in our representation, with the purpose to recognize the set of important domain-specific phrases, where the n-gram's importance is expressed by its frequency calculated as tf–idf weight.

Before computing the sentence representation with n-grams, we remove special symbols, phone numbers, emails and urls from a sentence. For a given maximum length L of an n-gram, we compute all of the existing n-grams for $n = 1$ to $n = L$ and calculate the tf–idf score for them as follows:

Let T be an n-gram with $|T|$ words in a document D_i having $|D_i|$ words in total, and let T appear DR_i times in the document D_i. Term frequency of T in a document D_i is calculated as

$$tf(T, D_i) = \frac{DR_i}{|D_i| - |T| + 1} \tag{1}$$

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

$$idf(T) = \log \frac{N}{CR} \tag{2}$$

The tf–idf score of an n-gram T in document D_i is:

$$tf\text{-}idf(T, D_i) = tf(T, D_i) \cdot idf(T) \tag{3}$$

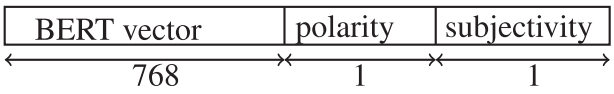

Fig. 2. Sentence representation for the BERT_{sent} model.

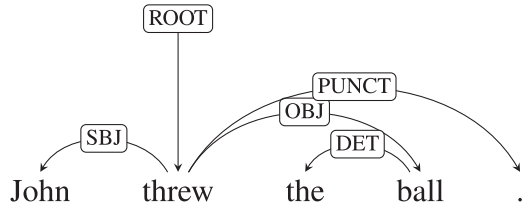

Fig. 3. Sentence dependency tree.

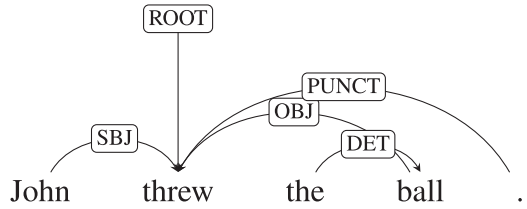

Fig. 4. Inverse sentence dependency tree.

Then, for a sentence S in a document D_i its tf–idf score is computed as

$$tf\text{-}idf(S) = \sum_{T \in S, 1 \le |T| \le L} tf\text{-}idf(T, D_i) \tag{4}$$

The n-grams that contain stopwords only[1] get zero tf–idf value.

3.4.2. Semantic sentence representation with BERT and sentiment

The semantic sentence representation, denoted by BERT_{sent}, concatenates BERT sentence vectors with sentence polarity and objectivity values, resulting in real-valued vectors of length 770. The shape of the data is depicted in Fig. 2.

3.4.3. Syntactic sentence representation with node embeddings

The syntactic sentence representation, denoted by DEP_{ne}, uses the sentence's dependency structure. We start with an ordinary dependency tree (an example is given in Fig. 3). First, we inverse the edges to obtain an in-tree with the sentence root as its sink; we do it because we want all of the random walks performed on that tree to halt at the root (an example is given in Fig. 4).

Node2Vec algorithm (Grover & Leskovec, 2016) generates a representation of the graph and its nodes as real vectors reflecting the network neighborhoods of nodes. The algorithm for the random walk generation traverses each node in the graph and generates a pre-defined number of random walks of a certain length. Both of these numbers are parameters that can be modified by the user. The probability of proceeding to an edge depends on an edge weight and a pre-defined transition probability, which is also a parameter. After all random walks have been generated, the Node2Vec algorithm treats them as 'sentences' in which nodes are 'words. Then, it applies a standard approach of generating word embeddings from sentences (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) to generate embeddings for the nodes.

We apply Node2Vec to the inverse dependency tree and generate a node embedding for every node in that tree. We extract the node embedding vector of the root node (Fig. 5) and concatenate it with the word vector of a root token (using the default pre-trained word vectors by Spacy (Honnibal & Johnson, 2015)).

We generated node embeddings of size 128, setting the number of walks to 10 and the walk length to 80 (the parameters were chosen empirically). Therefore, a sentence is represented by a 128-size vector of its root node. The final size of this representation is 228 (word vector of length 100 and node embedding of size 128), as shown in Fig. 6.

3.5. Sentence label prediction with semantic and syntactic representations

To turn semantic and syntactic sentence representations into weights, we treat the task of extractive summarization as a binary sentence classification task—a sentence is labeled 1 in a training set if it belongs to one of the gold summaries, and it is labeled 0 otherwise. These labels are then considered to be sentence weights.

We train **two separate prediction models**, one for the semantic sentence representation (Section 3.4.2), and one for the syntactic sentence representation (Section 3.4.3). For each of these representations, we train a Bidirectional LSTM neural model on the training data, and predict sentence labels for every sentence in the validation part of the FNS-2021 dataset. The network that we used has 50 neurons, and it was trained for 100 epochs (the parameters were chosen empirically). This classification pipeline is shown in Fig. 7.

To evaluate our networks and data representation, we used the truncated FNS-2021 shared task dataset as follows: we trained every model separately on the training set of this dataset and evaluated the validation set containing 363 files and 87,464 sentences. BERT_{sent} model achieved 0.962 accuracy and confusion matrix $(\text{TP}, \text{FP}, \text{FN}, \text{TN}) = (357, 1467, 1826, 83814)$. For the DEP_{ne} model the accuracy was 0.952 with confusion matrix as follows: $(\text{TP}, \text{FP}, \text{FN}, \text{TN}) = (1819, 26994, 364, 58287)$.

3.6. Summary generation and ranking

For every document D, we generate all the sequences of up to 1,000 words (there are $|D|-999$ such sequences in a document with more than 1,000 words). This procedure aimed to try and catch the cases where the human experts who generated the gold standard summaries have used the entire paragraphs or sections of the original document.[2]

For every such sequence Seq we calculate the following three parameters (shown in Fig. 8):

1. The min–max normalized sum of tf–idf values for all the n-grams of all the sentences S contained in this sequence:
$$tf\text{-}idf(Seq) = \sum_{S \in Seq} tf\text{-}idf(S) \tag{5}$$
2. The min–max normalized sum of semantic-based sentence labels produced by the BERT_{sent} prediction model for all the sentences in the sequence, denoted by $Score(Seq, \text{BERT}_{sent})$;
3. The min–max normalized sum of syntactic-based sentence labels produced by the DEP_{ne} prediction model for all the sentences in the sequence, denoted by $Score(Seq, \text{DEP}_{ne})$.

The final score of a sequence Seq is set to

$$\begin{aligned} Score(Seq) = \quad & \alpha \times tf\text{-}idf(Seq) + \beta \times Score(Seq, \text{BERT}_{sent}) \\ & + \gamma \times Score(Seq, \text{DEP}_{ne}) \\ \text{where} \quad & \alpha + \beta + \gamma = 1. \end{aligned} \tag{6}$$

We normalize this score by the number of sentences in a sequence and select the highest-ranking sequence as our summary. Given a pre-calculated dictionary of sentences with their scores, the calculation of scores for sentence sequences is highly time-efficient—around 3 min for 363 documents.

[1] We do not remove stopwords during the preprocessing, but examine every word against a custom stopword list as follows: ['and', 'the', 'is', 'are', ' this','at', 'of', 'to', 'in', 'on', 'for', 'or','a', 'an', 'as', 'page', 'by', 'with', 'our', 'we', 'that', 'may'].

[2] The gold-standard summaries of the FNS data are mainly extracts of entire sections.

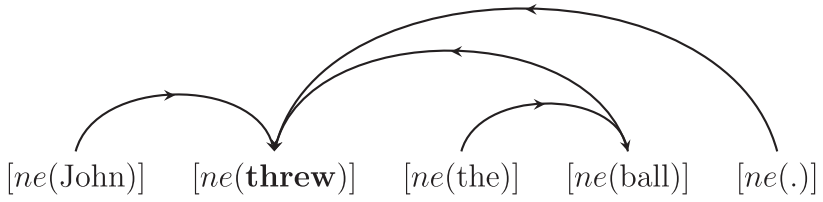

Fig. 5. Node embeddings generated for the inverse dependency tree (the root is marked in bold).

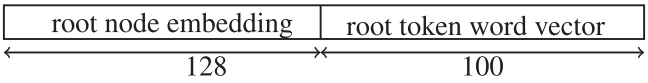

Fig. 6. Sentence representation for the DEP_{ne} model.

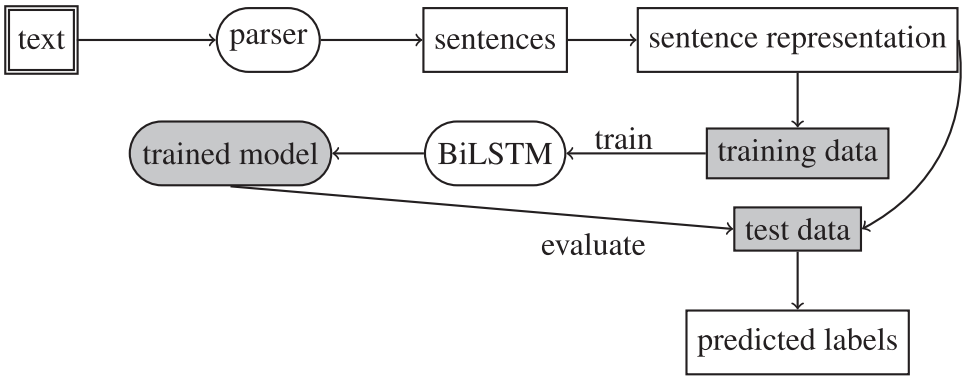

Fig. 7. Classification pipeline.

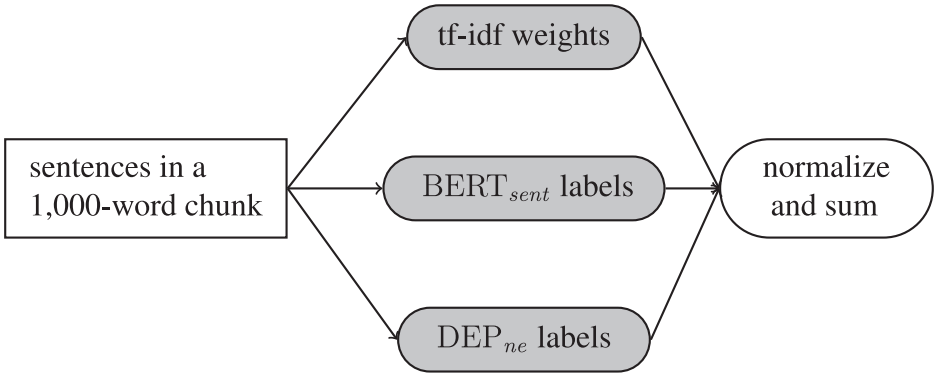

Fig. 8. Combined sentence weight.

4. Experiments

We performed an intrinsic evaluation for our summarization method and compared the summaries produced by our method to the gold standard summaries provided by the FNS 2021 shared task organizers.

4.1. Data description

The Financial Narrative Summarization (FNS 2021) shared task aims to demonstrate the value and challenges of applying automatic text summarization to financial text written in English, usually referred to as financial narrative disclosures. For the creation of the financial narrative summarization dataset, 3,863 UK annual reports published in PDF file format were used. UK annual reports are lengthy documents with around 80 pages on average, some annual reports could span more than 250 pages, while the summary length should not exceed 1,000 words. The training set includes 3,000 annual reports, with 3–4 human-generated summaries as a gold standard. For the evaluation and system development, the validation set of 363 files was provided. Because no gold standard summaries were provided with the test set, we performed our evaluations on the validation set. Table 1 contains the dataset statistics.

4.2. Tools and runtime environment

Experiments were performed on a cloud server with 32 GB of RAM, 150 GB of PAGE memory, an Intel Core I7-7500U 2.70 GHz CPU, and one NVIDIA GK210GL GPU.

Preprocessing was performed with the Spacy package (Honnibal & Johnson, 2015) v3.0. We use *en_core_web_sm* Spacy for a parsing, *en_core_web_trf* for BERT sentence embedding extraction, and for a sentiment analysis.

Table 1
FNS 2021 dataset statistics.

Dataset	# Documents	# Gold summaries	Avg sentences	Avg words	Avg chars
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

To generate node embeddings from sentence dependency structure, we used the networkX package (Hagberg, Swart, & S. Chult, 2008) to construct directed graphs; graph-based node embeddings for these trees were computed with the node2vec package (Grover & Leskovec, 2016).

4.3. Evaluation metrics

ROUGE (Lin, 2004) is a collection of metrics that are commonly used for evaluating automatic summarization in NLP. Originally, ROUGE was implemented as a Perl script, but numerous implementations in Java, Python and other languages are available. ROUGE compares an automatically produced summary generated by a system of a method to a reference or a set of *references summaries* that are usually produced by humans. Reference summaries are also often called *gold summaries*. ROUGE includes the following metrics:

- ROUGE-N for $N = 1, 2, \ldots$ that measures overlap of N-grams (Lin & Hovy, 2003) between the system summary and reference summaries. N-grams are consecutive sequences of words (or tokens) seen in the text. The most common ROUGE-N metrics are ROUGE-1 which measures the overlap of unigrams (single words), and ROUGE-2 which measures the overlap of bigrams (pairs of words) between the system and the reference summaries.
- ROUGE-L (L stands for Longest Common Subsequence (Lin & Och, 2004)) identifies the longest co-occurring sequence in sentence N-grams.
- ROUGE-W is a weighted variation of the Longest Common Subsequence metrics that gives higher weight to consecutive common sequences.
- ROUGE-S measures co-occurrence of skip-bigrams (Lin & Och, 2004) which are word pairs with gaps between them.
- ROUGE-SU searches for the co-occurrence of both the skip-bigrams and the unigrams.

Every ROUGE metric measures recall (fraction of relevant sequences that were retrieved), precision (fraction of relevant sequences among the retrieved sequences), and F-measure which is the harmonic mean of the two. In our evaluation we use all these ROUGE metrics.

4.4. Evaluated systems

All of our systems and baselines, described below, were applied to the validation part of the FNS-2021 shared task dataset, and results are reported in Tables 2 and 3.

4.4.1. Baselines

We compared our results to two baselines: a trivial TOP-K baseline that includes the first 1,000 words of a document, and MUSE (Litvak et al., 2010)—a supervised sentence extractor based on the Genetic Algorithm—that was used in both FNS shared tasks as a topline.

4.4.2. Methods combining tf–idf, syntactic and semantic information

We evaluated the following setups of our approach:

- n-grams vectors with BERT_{sent} labels, given $\alpha = \beta = \frac{1}{2}$ and $\gamma = 0$ in (6) (denoted as NG_L+BERT),
- n-gram vectors with DEP_{ne} labels, given $\alpha = \gamma = \frac{1}{2}$ and $\beta = 0$ in (6) (denoted by NG_L+DEP),
- n-grams vectors with both labels, given $\alpha = \beta = \gamma = \frac{1}{3}$ in (6) (denoted by NG_L+BERT+DEP),

Table 2
Results for FNS-2021 validation set. Rouge-1 and Rouge-2.

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	0.422	0.400	0.397	0.262	0.182	0.204
NG_1	0.433	0.393	0.398	0.270	0.175	0.202
NG_2	0.444	0.394	0.403	0.281	0.177	0.207
NG_3	**0.448**	0.393	0.405	**0.285**	0.177	0.208
NG_1+BERT	0.422	0.400	0.397	0.262	0.182	0.204
NG_2+BERT	0.424	0.401	0.399	0.265	0.183	0.206
NG_3+BERT	0.425	0.402	0.399	0.265	0.183	0.206
NG_1+DEP	0.429	0.393	0.396	0.267	0.174	0.200
NG_2+DEP	0.440	0.392	0.401	0.277	0.175	0.205
NG_3+DEP	0.443	0.391	0.402	0.280	0.175	0.206
NG_1+BERT+DEP	0.422	0.401	0.398	0.261	0.181	0.204
NG_2+BERT+DEP	0.424	0.401	0.399	0.264	0.182	0.206
NG_3+BERT+DEP	0.425	0.402	0.400	0.265	0.183	0.206
BERTSUM	0.433	0.404	0.404	0.270	0.196	0.216
NG_1+BERT+DEP+BERTSUM	0.439	0.410	**0.410**	0.277	0.198	**0.220**
NG_2+BERT+DEP+BERTSUM	0.437	0.409	0.409	0.276	0.197	0.219
NG_3+BERT+DEP+BERTSUM	0.438	0.409	0.409	0.277	0.197	0.219
BERT+BERTSUM	0.429	**0.412**	0.406	0.269	**0.200**	0.218
NG_3+BERTSUM	0.442	0.399	0.406	0.279	0.191	0.216

Table 3
Results for FNS-2021 validation set. Rouge-L and Rouge-SU4.

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	0.410	0.372	0.380	0.325	0.177	0.219
NG_1	0.412	0.368	0.379	0.335	0.172	0.218
NG_2	0.419	0.372	0.384	0.346	0.174	0.222
NG_3	0.422	0.372	0.386	**0.350**	0.174	0.223
NG_1+BERT	0.410	0.372	0.380	0.325	0.177	0.219
NG_2+BERT	0.411	0.373	0.382	0.328	0.177	0.221
NG_3+BERT	0.413	0.373	0.383	0.328	0.178	0.221
NG_1+DEP	0.410	0.368	0.378	0.331	0.171	0.217
NG_2+DEP	0.416	0.370	0.382	0.342	0.173	0.221
NG_3+DEP	0.418	0.371	0.384	0.345	0.173	0.221
NG_1+BERT+DEP	0.410	0.372	0.381	0.325	0.176	0.219
NG_2+BERT+DEP	0.411	0.373	0.382	0.327	0.177	0.221
NG_3+BERT+DEP	0.413	0.374	0.383	0.328	0.178	0.221
BERTSUM	0.409	0.381	0.384	0.333	0.196	0.236
NG_1+BERT+DEP+BERTSUM	**0.423**	**0.384**	**0.393**	0.339	0.197	**0.239**
NG_2+BERT+DEP+BERTSUM	0.421	**0.384**	0.392	0.338	0.197	**0.239**
NG_3+BERT+DEP+BERTSUM	0.422	**0.384**	0.392	0.338	0.197	**0.239**
BERT+BERTSUM	0.416	0.383	0.389	0.330	**0.199**	0.238
NG_3+BERTSUM	0.416	0.379	0.387	0.342	0.191	0.236

- pure n-grams vectors, with $\alpha = 1$ and $\beta = \gamma = 0$ in (6) (denoted by NG_L).

Also, we experimented with

- $L = 1$ (only unigrams which is equivalent to the classic bag-of-words),
- $L = 2$ (unigrams and bigrams), and
- $L = 3$ (unigrams, bigrams, and trigrams).

In total, we got 12 different setups, denoted as NG_1+BERT, NG_2+BERT, NG_3+BERT, etc.

4.4.3. Adding BERT-based ranking

We also experimented with a simple ranking approach based on BERT sentence vectors (denoted BERTSUM), where a sentence score is a sum of vector values and a sequence score $Score(Seq)$ is a sum of

Table 4
Runtime statistics.

System	Full validation set	Validation cut to 10%
TIBER with $L = 1$	84 s	34 s
TIBER with $L = 2$	176 s	46 s
TIBER with $L = 3$	321 s	68 s
MUSE	19,200 s	255 s

scores for all its sentences. We followed an intuition where semantic features probably reflect the importance of a sentence to the "topics" represented by these features.

Because we observed unexpectedly high performance of this simple approach, we also decided to check its combinations with other setups, namely:

- tf–idf vectors over n-grams and labels produced by BERT_{sent} and DEP_{ne} models (denoted by NG_L+BERT+DEP+BERTSUM);
- only BERT_{sent} labels (denoted by BERT+BERTSUM), and
- only tf–idf vectors over n-grams (only for $L = 3$ as the best performing among all NG_L representations; and denoted by NG_3+BERTSUM).

4.5. Evaluation results and discussion

We applied four Rouge (Lin, 2004) metrics—Rouge-1, Rouge-2, Rouge-L, and Rouge-SU4 on the validation set. We used ROUGE 2.0 (Ganesan, 2015) package[3] with its default list of stopwords.[4]

Tables 2 and 3 show the results, with recall, precision, and F-measure for all Rouge metrics.

As can be observed, n-grams with maximal length $L = 3$ gain a recognizable advantage over smaller n-grams in terms of all Rouge metrics. This outcome demonstrates the importance of the presence of financial terminology in the generated summaries. However, a simple NG_3 usually produces a high Recall but not Precision.

Another observation is that BERT vectors gain much more valuable semantic information than node vectors obtained on syntactic trees. We can conclude that the semantics of a sentence is a much better indicator of its importance for summarization than its syntactic structure.

It can be seen that our best methods significantly outperform the topline (MUSE) in all metrics. If we compare the different setups, we can see that the combination of all four components (NG_1+BERT+DEP+BERTSUM) outperforms other methods in terms of F-measure of all Rouge metrics.

Table 4 compares between runtimes of TIBER and MUSE. The most "heavy" TIBER's options, which include tf–idf, syntactic, and semantic information, are limited by the numbers shown in Table 4 which are much lower than MUSE's runtimes.

5. Limitations of our study

Our method is currently limited to English texts, mainly due to its dependency on the syntactic parser and other external tools. In general, our method can be extended to additional languages if a preprocessing tool, dependency parser, semantic vectors, and sentiment analyzer are provided for that languages.

Also, our method was evaluated on the FNS-2021 dataset only, due to limited sources on financial summarization. This dataset is known to have the following limitations: the documents are extremely long, contain numerous tables, charts, and numerical data, and use the "financial" language with vocabulary different from ones of other domains.

To summarize lengthy reports in a reasonable time, while focusing on their most important parts, our method only processes the first 10% of each file in the training and validation sets of the FNS-2021 dataset. Despite it comes in line with the insight of Gokhan et al. (2021) and our observation that most of the gold summaries for the training set of FNS-2021 are contained in the first 10% of a document, this limitation creates a serious bias.

6. Conclusions and future work

This paper introduces a new summarizer for financial reports. The method combines several techniques, such as n-grams, BERT, sentiment analysis, and node embedding, for building a rich sentence representation and further ranking of consequent sentence sequences. Our ranking procedure is simple, straightforward, and very time-efficient. The evaluation results show that (1) semantic information is more critical for selection summary sentences than syntactic information; (2) the presence of financial terminology—represented by n-grams with $1 \leq n \leq 3$—in the generated summaries is important; and (3) our best method significantly outperforms the topline. Also, our results support the observation made by Gokhan et al. (2021) that preliminary filtering of 90% of an input text filters out irrelevant information and improves the computational complexity and accuracy of the summarization procedure, including the application of neural models.

Despite one may expect from a highly accurate system that it will be computationally expensive, our approach is very time-efficient. For generating node embeddings, we do not represent the entire document as a graph, but only a sentence. Only classification models, predicting sentence labels for a joint representation, need training data. Once these models are pre-trained, they can be applied to new sentences in a very efficient manner. Also, preliminary selection of the first 10% of the original reports both reduces computational resources and eliminates irrelevant information.

The future work may include exploring transformer-based models that are designed to process long sequences such as Longformer (Beltagy, Peters, & Cohan, 2020).

CRediT authorship contribution statement

Natalia Vanetik: Conceptualization, Methodology, Software, Writing. **Marina Litvak:** Methodology, Writing, Project administration. **Sophie Krimberg:** Software, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agarwal, R., Verma, I., & Chatterjee, N. (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* (pp. 33–39).

Agrawal, Y., Anand, V., Arunachalam, S., & Varma, V. (2021). Hierarchical model for goal guided summarization of annual financial reports. In *Companion proceedings of the web conference 2021* (pp. 247–254).

Agrawal, Y., Anand, V., Gupta, M., Arunachalam, S., & Varma, V. (2021). Goal-directed extractive summarization of financial reports. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 2817–2821).

Arora, P., & Radhakrishnan, P. (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* (pp. 137–142).

[3] https://github.com/kavgan/ROUGE-2.0.

[4] Rouge scores reported by FNS2021 organizers were generated using different Rouge setup—including stopwords—and therefore cannot be compared to ours. Including stopwords in Rouge comparisons between texts significantly raise its scores. For example, Rouge-1 Precision for MUSE raises to 0.52 when we include stopwords.

Azzi, A. A., & Kang, J. (2020). Extractive summarization system for annual reports. In *Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation* (pp. 143–147).

Baralis, E., Cagliero, L., & Cerquitelli, T. (2016). Supporting stock trading in multiple foreign markets: a multilingual news summarization approach. In *Proceedings of the second international workshop on data science for macro-modeling* (pp. 1–6).

Baviskar, D., Ahirrao, S., & Kotecha, K. (2021). Multi-layout unstructured invoice documents dataset: A dataset for template-free invoice processing and its evaluation using AI approaches. *IEEE Access*, *9*, 101494–101512.

Baviskar, D., Ahirrao, S., Potdar, V., & Kotecha, K. (2021). Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions. *IEEE Access*.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Bragg, S. (2021). Accounting CPE courses and books. https://www.accountingtools.com. (Accessed: 2021-11-27).

de Oliveira, P. C. F., Ahmad, K., & Gillam, L. (2002). A financial news summarization system based on lexical cohesion. In *Proceedings of the international conference on terminology and knowledge engineering, Nancy, France*.

El-Haj, M. (2019). MultiLing 2019: FInancial narrative summarisation. In *Proceedings of the workshop multiling 2019: summarization across languages, genres and sources* (pp. 6–10).

El-Haj, M., Athanasakou, V., Ferradans, S., Salzedo, C., Elhag, A., Bouamor, H., et al. (2020). 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*.

El-Haj, M., Litvak, M., Pittaras, N., Giannakopoulos, G., et al. (2020). The financial narrative summarisation shared task (FNS 2020). In *Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation*, (pp. 1–12).

El-Haj, M., Rayson, P., & Moore, A. (2018). The first financial narrative processing workshop (FNP 2018). In *Proceedings of the LREC 2018 workshop*.

Filippova, K., Surdeanu, M., Ciaramita, M., & Zaragoza, H. (2009). Company-oriented extractive summarization of financial news. In *Proceedings of the 12th conference of the european chapter of the ACL (EACL 2009)* (pp. 246–254).

Ganesan, K. (2015). ROUGE 2.0: UPdated and improved measures for evaluation of summarization tasks.

Gokhan, T., Smith, P., & Lee, M. (2021). Extractive financial narrative summarisation using SentenceBERT based clustering. In *Proceedings of the 3rd financial narrative processing workshop* (pp. 94–98).

Grover, A., & Leskovec, J. (2016). node2vec: SCalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864).

Hagberg, A., Swart, P., & S. Chult, D. (2008). *Exploring network structure, dynamics, and function using networkx*: *Technical report*, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Honnibal, M., & Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1373–1378). Lisbon, Portugal: Association for Computational Linguistics, URL https://spacy.io/.

Isonuma, M., Fujino, T., Mori, J., Matsuo, Y., & Sakata, I. (2017). Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2101–2110).

Krimberg, S., Vanetik, N., & Litvak, M. (2021). Summarization of financial documents with TF-IDF weighting of multi-word terms. In *Proceedings of the 3rd financial narrative processing workshop* (pp. 75–80).

La Quatra, M., & Cagliero, L. (2020). End-to-end training for financial report summarization. In *Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation* (pp. 118–123).

Li, L., Jiang, Y., & Liu, Y. (2020). Extractive financial narrative summarisation based on DPPs. In *Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation* (pp. 100–104).

Li, S., Shi, W., Wang, J., & Zhou, H. (2021). A deep learning-based approach to constructing a domain sentiment lexicon: a case study in financial distress prediction. *Information Processing & Management*, *58*(5), Article 102673.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the north american chapter of the association for computational linguistics*, (pp. 150–157).

Lin, C.-Y., & Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, (pp. 605–612).

Litvak, M., Last, M., & Friedman, M. (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* (pp. 927–936).

Litvak, M., & Vanetik, N. (2021). Summarization of financial reports with AMUSE. In *Proceedings of the 3rd financial narrative processing workshop* (pp. 31–36).

Litvak, M., Vanetik, N., & Puchinsky, Z. (2020). SCE-SUMMARY At the FNS 2020 shared task. In *Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation* (pp. 124–129).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, *26*.

Orzhenovskii, M. (2021). T5-LONG-EXTRACT at FNS-2021 shared task. In *Proceedings of the 3rd financial narrative processing workshop* (pp. 67–69).

Singh, A. (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* (pp. 105–111).

Suarez, J. B., Martínez, P., & Martínez, J. L. (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* (pp. 112–117).

Vhatkar, A., Bhattacharyya, P., & Arya, K. (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* (pp. 130–136).

Yang, C. C., & Wang, F. L. (2003). Automatic summarization for financial news delivery on mobile devices. In *WWW (Posters)*.

Zhang, Y., Chen, E., & Xiao, W. (2018). Extractive-abstractive summarization with pointer and coverage mechanism. In *Proceedings of 2018 international conference on big data technologies* (pp. 69–74).

Zheng, S., Lu, A., & Cardie, C. (2020). SUMSUM@ FNS-2020 shared task. In *Proceedings of the 1st joint workshop on financial narrative processing and multiling financial summarisation* (pp. 148–152).

Zmandar, N., El-Haj, M., Rayson, P., Litvak, M., Giannakopoulos, G., Pittaras, N., et al. (2021). The financial narrative summarisation shared task FNS 2021. In *Proceedings of the 3rd financial narrative processing workshop* (pp. 120–125).

Zmandar, N., Singh, A., El-Haj, M., & Rayson, P. (2021). Joint abstractive and extractive method for long financial document summarization. In *Proceedings of the 3rd financial narrative processing workshop* (pp. 99–105).