EXTRACTIVE SUMMARISATION OF UK ANNUAL REPORTS

A REPORT SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF BACHELOR OF SCIENCE
IN THE FACULTY OF SCIENCE AND ENGINEERING

Vladislav Yotkov

Supervised by: **Dr. Jonathan Shapiro**

Department of Computer Science

April 2023

Contents

Word Count: 12104

List of Tables

List of Figures

Acronyms

AF	Accounting and Finance
ATS	Automatic Text Summarisation
BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag of Words
CSF	Computational Shared Facility
ESG	Environmental, Social, and Governance
ESMA	European Securities and Markets Authority
ETS	Extractive Text Summarisation
FCFFNN	Fully-Connected Feed-Forward Neural Network
FinBERT	Financial Bidirectional Encoder Representations from Transformers
FNP	Financial Narrative Processing
FNP21	Financial Narrative Processing 2021
FNP22	Financial Narrative Processing 2022
FNS21	Financial Narrative Summarisation 2021
FNS22	Financial Narrative Summarisation 2022
FRC	Financial Reporting Council

GRU	Gated Recurrent Unit
IFRS	International Financial Reporting Standards
LCS	Longest Common Subsequence
LED	Longformer-Encoder-Decoder
LSTM	Long Short-Term Memory
MLM	Masked Language Model
NLP	Natural Language Processing
NSP	Next Sentence Prediction
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SEC	Securities and Exchange Commission
Tf-Idf	Term Frequency - Inverse Document Frequency

Abstract

Although there has been considerable progress in Natural Language Processing (NLP) over the years, it has not fully reached the Accounting and Finance (AF) industry. In the meantime, companies worldwide produce vast amounts of textual data as part of their reporting packages to comply with regulations and inform shareholders of their financial performance. The glossy annual report is such an example, widely read by investors, which also tends to be quite long. Inspired by the Financial Narrative Summarisation (FNS) workshops [ZEHR+21, EHRZ22], we design an Automatic Text Summarisation (ATS) system for the narrative parts of UK financial annual reports. With this goal in mind, we implement and explore the following models for Extractive Text Summarisation (ETS): *attention-based Financial Recurrent Neural Network (FinRNN) with data augmentation*, and *fine-tuned Financial BERT (FinBERT)* [YUH20]. Our evaluations based on the ROUGE-2 metric show both models to be outperforming the standard ATS baselines: TextRank [MT04], and LexRank [ER04]. Furthermore, our proposed FinBERT-base demonstrates competitive performance against official FNS 2022 models on the validation set - achieving an average ROUGE-2 $F1$ score of 0.382 beating with 0.081 the *best performing model overall in the FNS22 competition* - the *mT5* [FRM+22].

Declaration

No portion of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I would like to thank my supervisor Dr. Jonathan Shapiro and Dr. Riza Batista-Navarro for the provided advice along the way. Also, this project could not have been possible without the UK annual reports data and the computing resources provided by the Financial Narrative Processing's administration and the UoM's Computational Shared Facility, respectively.

Introduction

> "Son," my father said to me, "someday this will all be yours."
>
> Kurt Vonnegut, Jr.

In this chapter, we explain what financial reports are and specifically focus on UK annual reports. We outline the main the challenges the latter present for large-scale Natural Language Processing (NLP) research. Furthermore, we show that the Accounting & Finance (AF) research and industry have not adopted the latest NLP techniques and we introduce the Financial Narrative Processing (FNP) workshops as a response to this problem. Finally, we state our aims and objectives for this third-year project before we outline the background and related work in Chapter 2.

1.1 Financial Reports

Due to international regulations, companies are obliged to report their periodic performance (annual, bi-annual, quarterly) to various regulatory authorities[1] and other users (e.g., corporate stakeholders, investors, customers, suppliers, etc.). These reports contain essential information about

[1] Regulation authorities worldwide:

- Securities and Exchange Commission (SEC) in the USA
- European Securities and Markets Authority (ESMA) in Europe
- Financial Reporting Council (FRC) in the UK
- International Financial Reporting Standards (IFRS) in 167 jurisdictions worldwide

the operations and finances of a business and are crucial for making informed decisions (from a user perspective), but are different in regulatory forms. For example,

1. 10-K reports filed to the SEC[2] and accessible through their Electronic Data Gathering, Analysis, and Retrieval[3] (EDGAR) system are only for US registered businesses. They follow a standardised template and are plain text, which makes them particularly easy for automated large-scale research [EHAR+19]. Also, the contents of these reports are strict, requiring solely five information sections[4].

2. UK annual reports, regulated by the UK's Financial Reporting Council (FRC), are typically the primary annual reporting method (also provided as PDF files). Unlike the 10-K, they are glossy and more stakeholder-oriented and enjoy unlimited discretion over non-mandated content [EHAR+19] (e.g., photography and company brand material, non-mandatory narrative sections, etc.). However, these are more challenging for automated processing due to their variable section structure, formatting, and rich visual representations.

1.2 UK annual reports

The annual report is the primary corporate disclosure legally required for public companies by regulatory authorities. While it *does not have a rigid document structure* like the 10-K, it typically has a *narrative component*[5] and the financial statements (at the rear).

As we outlined in Section 1.1, UK annual reports have the following inconvenient properties with regard to large-scale text understanding (see example excerpts from Oxfam's annual report provided in the Appendix - Figures 6.1 and 6.2).

- They are very long documents. Throughout the years, their average length has been increasing significantly with the number of pages rising 57% for the median report from 2003 to 2016 (47 to 74 pages, respectively) [LY19], due to additional regulations between 2006 and 2008 ([EHAR+19];

[2]https://www.sec.gov

[3]https://www.sec.gov/edgar

[4](a) Business Overview (b) Risk Factors (c) Management's Discussion and Analysis of Financial Condition and Results of Operations (MD&A) (d) Financial Statements (e) Supplementary Disclosures

[5]The narrative component of a UK annual report typically consists of 1. Management's Commentary 2. Letter to Shareholders 3. Corporate Governance Statement 4. Auditor's Report 5. Remuneration Report 6. Business Review 7. Environmental, Social, and Governance (ESG) Report 8. Risk Management Report

- They have variable nomenclature. From firm to firm, naming conventions vary "dramatically", with more than 20 unique titles for various sections (e.g., Chair's letter to shareholders, Management Commentary) [LY19];

- They incorporate embedded info-graphics. While domain experts hail the integration of highly interactive elements into corporate reporting [KB16], the compilation to PDF makes the task of analysing such unstructured documents automatically even harder [LY19];

These challenges motivate the work of [EHRY+19] who (a) established a set of 8 generic section headers[6] and (b) built the CFIE-FRSE[7] extraction tool that converts a text-based PDF annual report to simple text.

1.3 NLP in Accounting and Finance

The relevance of this project should also be understood from the perspective of the development of Natural Language Processing (NLP) in the Accounting and Finance (AF) domain. As outlined in [Ell98], investors' trust in the accountability of businesses would be based no longer as much on just the financial statements, but also on more descriptive narratives that define strategy and planning of resource use. While some recognise the importance of understanding in-domain textual information [L+10], others like [EHRW+19] report that the industry is still doubtful and cynical about the NLP applications in the analysis of financial market disclosures. Furthermore, the latter also observe that AF researchers rely extensively on bag–of–words models, which are *not sufficient to encode complex contextual and semantic meaning* (especially in a domain with such *specialized language*). As for ATS [CHW19] is said to be the single AF study into disclosure summarisation. Its authors demonstrate that machine-generated summaries are less likely to bias positively investor decisions compared to managerial ones. Therefore, this only confirms the existence of a wide gap in NLP applications in Accounting research, which further motivates our work.

[6] (a) Chairman Statement (b) CEO Review (c) Corporate Governance Report (d) Directors Remuneration Report (e) Business Review (f) Financial Review (g) Operating Review (h) Highlights

[7] The CFIE-FRSE stands for Corporate Financial Information Environment - Final Report Structure Extractor. It is publicly available at `https://github.com/drelhaj/CFIE-FRSE` and it can be used to convert English, Spanish and Portuguese annual reports.

1.4 Financial Narrative Summarisation (FNS) Task

The FNS Task is part of the annual Financial Narrative Processing (FNP) Workshop [8] organised by Lancaster University since 2018, which aims to:

- encourage the advancement of financial text mining & narrative processing
- examine methods of structured content retrieval from financial reports
- explore causes and consequences of corporate disclosure

as stated in their inaugural proceedings [9].

For that purpose, they produce datasets of extracted narratives (with the help of the CFIE-FRSE tool) from annual reports of UK companies listed on the London Stock Exchange (LSE).

In their FNS 2022 Task, there were $3,863$ such reports in English (Table 1.1), while the average length was reported at 80 pages, and the maximum of more than 250 pages [LV21].

Additionally, for every report, there were at least two gold summaries situated in the annual report itself [10] The workshop's goal was to build ATS systems that generate a single summary for an annual report, no longer than $1,000$ words (shorter than the gold summaries on average).

Data Type	Training	Validation	Testing	Total
Report full text	3,000	363	500	3,863
Gold summaries	9,873	1,250	1,673	12,796

Table 1.1: FNS22 Data Split [EHZR$^+$22]

We acknowledge that due to the scarcity of publicly available financial data this third-year project could not have been possible without the kind permission of the FNP organisers to use the training and validation datasets from their FNS22 Task [EHRZ22].

[8] `https://wp.lancs.ac.uk/cfie/`

[9] `https://wp.lancs.ac.uk/cfie/fnp2018/`

[10] The gold summaries being already in the annual report is not problematic because these reports are already written by domain experts who know how to summarise the financial state of a company. Hence, multiple sections/paragraphs could achieve this thoroughly, and the organisers have identified & extracted them manually with the help of the professional writers of the individual reports. At this moment, one can begin to doubt the point of applying ATS techniques, but due to the *lack of rigid document structure*, *it is not trivial to automatically find these text excerpts with heuristic methods*. Furthermore, we can formulate this challenge as finding the latent features of a summarising (i.e., "to-be-in-the-summary") sentence, highlighted as one of the fundamental advantages of NLP in AF research ([LY19], [EHRW$^+$19]).

1.5 Aim and Objectives

The summarisation of UK annual reports is a challenging task because of:

- the various inconveniences of the reports around their large-scale understanding (Section 1.2);
- the discrepancy between Accounting and Finance (AF) research in NLP and the general NLP field (Section 1.3);
- the nature of long-text summarisation in terms of available training data, financial language representation, complex model architectures, and reliability of evaluation metrics (Chapter 2);

Nevertheless, we decide to take up this challenge, being motivated by recent activities in the Financial NLP field (Section 1.4), and design Extractive Summarisation Models that perform better than the established baselines: TextRank [MT04], and LexRank [ER04]. For that purpose, several objectives had to be made:

1. pre-processing noisy report narratives (Section 3.1) and transforming them into suitable datasets for extractive summarisation (Section 3.2);
2. researching and incorporating the public financial state-of-the-art word embeddings (Section 2.4) for an effective text representation;
3. building an extractive neural model (Section 3.4) and tuning its hyperparameters for optimal classification capabilities (Section 3.7);
4. researching approaches on dealing with imbalanced datasets and implementing a data augmentation technique for a more discriminative learning process (Section 3.3);
5. exploring the capabilities of a pre-trained financial transformer (Section 3.5);
6. evaluating all summarisation models with the help of the FNS metric - ROUGE-2 (Section 4.2);

1.6 Project Structure

The project report is comprised of five chapters:

1. Chapter 1 outlines the background for UK annual reports and states our aim and objectives.
2. Chapter 2 provides the necessary background information to comprehend the problem of text summarisation, including the related work, and the evaluation metrics.
3. Chapter 3 outlines the methodology of the project, including the description of the data, the specifications of the models, and their hyperparameter tuning.
4. Chapter 4 presents our model results, including a quantitative and a qualitative evaluation of the produced output.
5. Chapter 5 concludes the project, summarising the main achievements and innovations, while also discussing a direction for future work.

Background

In this chapter we provide the necessary background knowledge for the reader to understand our proposed solution in Chapter 3. We will describe key concepts in modern extractive summarisation, such as word embeddings, Recurrent Neural Networks, attention mechanism and Transformer architecture. We also present various official FNS and baseline models which we use as benchmarks in our evaluation phase (Section 4.2).

2.1 Supervised Learning

Supervised learning is a machine learning paradigm where a training set X with data points x is provided, along with the corresponding set Y with labels y. Then the goal is to learn a function f that maps X to Y, i.e., $f : X \to Y$, which makes predictions $f(x)$ as close as possible to the true labels y [SW11]. The difference between $f(x)$ and y is quantified by a loss function L, which is being minimized during the training process by updating the model parameters with respect to the gradient of the loss [GBC16]. In the context of extractive summarisation, supervised learning can be used to identify the most important sentences in a document, which can then be used to generate a summary. In this case, the task can be cast as a binary classification problem, where the sentence labels are either 1 (summarising) or 0 (non-summarising) [MRS08].

2.2 Unsupervised Learning

Unsupervised learning is a machine learning paradigm where no labels are provided.

2.2.1 K-Means Clustering

K-Means clustering [Mac67] is an unsupervised learning algorithm that partitions a set of data points into k clusters, where each data point belongs to the cluster with the nearest mean. It is extensively used in extractive text summarisation for clustering of sentence embeddings [GSL21, FRM$^+$22].

2.3 Tf-Idf

Tf-Idf (Term Frequency - Inverse Document Frequency) is a statistical technique intended to reflect the importance of a word to a document in a corpus. It is often used as a weighting factor in information retrieval and text mining. The term frequency (Eq.2.1) is the number of occurrences of term t in document d [Luh57], but it is not enough to capture the importance of a term due to *all terms being considered equally important* [MRS08]. Meanwhile, the *document frequency (df_t)* is the number of documents in the corpus containing the term t, which evaluates how *common* and *unimportant* a term is [LRU20]. Therefore, *idf* becomes $\frac{|D|}{\mathrm{df}_t}$, where $|D|$ is the corpus size. It is also sometimes written in log-form as in Equation 2.2 due to the $|D|$ being large. Then the tf-idf term weight $w_{t,d}$ is calculated as the product of the two elements (Eq.2.3) [JM00]. This formulation now represents the importance of a term t normalised by its *commonness*. The tf-idf has also been successfully integrated in summarisation methods like LexRank [ER04] (Section 2.12) and in FNS competitive systems [KVL21, EHO22].

$$\mathrm{tf}(t,d) = \log\left(\mathrm{count}(t,d) + 1\right) \tag{2.1}$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2.2}$$

$$\text{tf-idf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D) \tag{2.3}$$

where t is a term, d is a document within a collection of documents/sentences D.

2.4 Word Embeddings

Historically, to represent a token (i.e., word) w_i in a vocabulary V numerically, we define a one-hot-encoding vector of all zeroes except of a one at the index of the word w_i in V (i.e., i).

The results are sparse individual word vectors being orthogonal to each other which 1. waste memory (each word is a $|V|$-sized vector, hence a total of $|V|^2$ for all tokens) and more importantly 2. fail to encode semantic similarity due to their cosine similarity being always zero.

Traditionally, AF research has represented an input text with the help of bag-of-words (BOW) models which can be viewed from the 1. the binary perspective - represent a whole document d as a binary vector containing ones for all words w_i occurring in d from V, 2. the term frequency perspective - encode number of word occurrences in documents instead of binary representation [XCWS13], and 3. the tf-idf perspective - extend the latter to penalise ubiquitous terms [Jon72]. Nevertheless, these vectors are very sparse and unable to encode more complex contextual and semantic meaning.

To address these shortcomings *short*[1] and *dense*[2] word embeddings like Word2Vec [MCCD13] and FastText [BGJM17] have been developed. In [MCCD13] the authors manage to condense the vector space and ensure that word representations have *multiple degrees of similarity* [MYZ13] (e.g., semantic - the meaning of words, morphological - structure of sub-words, etc).

Furthermore, the proposed models - CBOW (Continuous Bag of Words[3]) and Skip-gram evidently capture subtle semantic relationships and allow intuitive arithmetic operations as shown in the popular analogy: $\overrightarrow{\text{king}} - \overrightarrow{\text{man}} + \overrightarrow{\text{woman}} \approx \overrightarrow{\text{queen}}$[4].

The intuition for the two Word2Vec models is that in CBOW, the context (i.e., the surrounding tokens) is used to predict the middle token, while in skip-gram, the input token is used to predict the context (i.e., the surrounding tokens) (Figure 2.1).

Meanwhile, internally, the context prediction is cast as a binary classification task with positive examples being the target word and its surroundings, whereas the negatives ones are generated through random sampling from the dictionary. Then, the CBOW embeddings are the learned

[1] i.e., with a small number of dimensions

[2] i.e., continuous real-numbered values instead of 0/1s

[3] CBOW naming is derived from 1. the continuous distributed representation of the context and 2. the projection layer being shared across context words, i.e., the order of words does not affect the projection (similar to how bag-of–words model fails to encode word order).

[4] For a formal explanation on how analogies are realised in word embeddings we direct the readers to [EDH19]

weights of a logistic regression classifier with future and history words (i.e., the context window) as the input and the goal of correctly classifying the word in-between. In contrast, the Skip-gram uses the middle word as an input to the classifier and predicts the individual context words around it.

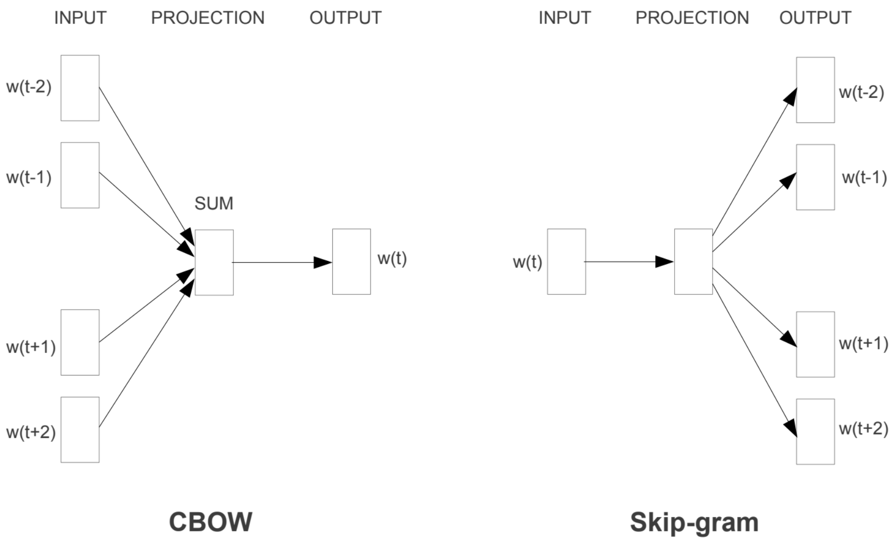


Figure 2.1: Word-from-context and context-from-word prediction in CBOW and Skip-gram, respectively, taken from [MCCD13]

A downside to the Word2Vec models is that they cannot handle out-of-vocabulary (OOV) word tokens, i.e., they cannot generate an embedding vector for words missing from the training data, which is crucial in real-life problems with *noisy* input or morphologically rich languages. For this reason [BGJM17], propose FastText as an extension to the Skip-gram model that makes use of character-level information to deal with unknown tokens. Here, each word is itself a bag of character n-grams which captures meaning of prefixes, suffixes, and morphemes. Additionally, two symbols are further introduced to mark the beginning and the end of a token, and help differentiate between sub-words and short words. For example, the character trigrams of the word *believe* are <*be, bel, eli, lie, iev, eve, ve*> where the sub-word *lie* is different from the word token <*lie*>.

Therefore, the final target word embedding is the sum of its constituent character n-grams which are learned via the Skip-gram model. This makes FastText very convenient for representing unknown words as the sum of *static* constituent n-grams [JM00].

Nevertheless, when it comes to domain-specific problems, general pre-trained word embeddings do not perform very well [RZP21]. demonstrate that even state-of-the-art embedding models like Google's Word2Vec(skip-gram)[5] and Facebook's FastText(skip-gram)[6]trained on 100 billion and 16 billion words, respectively, struggle to understand financial language like 1. *apple* standing for the company *Apple*, 2. ticker analogies, e.g., *amazon* is to *X* as *microsoft* is to *msft*, 3. grouping company name to ticker, exchange and country. In the same paper, the authors propose using the same algorithms (i.e., CBOW, Skip-gram, and FastText) but training solely on financial data instead - 15 years of financial news from the Dow Jones Newswires Text News Feed database, to produce the FinText models. They report a substantial increase in performance in and sensitivity to detecting financial jargon and relationships. In our project we acknowledge that a purpose-built financial word embedding (trained on proprietary data) will be more beneficial and more suited for the task of text summarisation of annual financial reports, which is why we select FinText as our preferred model.

2.5 Recurrent Neural Networks (RNNs)

The vanilla RNN is a basic type of RNN architecture designed for processing sequential data. It learns temporal patterns from the initial data by looping over the hidden layers which allow information to persist (i.e., they serve as a network memory) [Ola15]. The key component is the recurrent hidden state h_i (Figure 2.2a) updated at each time step using input data and the previous hidden state (Eq.2.4). This allows the RNN to capture contextual information and temporal dependencies in the sequence. However, due to the inherent vanishing and exploding gradient problems with the vanilla RNNs, they have limited ability to learn long-term dependencies [BSF94]. To resolve these issues more advanced RNN architectures like LSTMs and GRUs have been developed.

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h)^7 \tag{2.4}$$

[5]https://code.google.com/archive/p/word2vec/

[6]https://fasttext.cc/

[7]The algebraic formulation of the vanilla RNN has the following variables: 1. x_t is the input at time step t. 2. h_{t-1} is the hidden state at time step $t-1$. 3. h_t is the hidden state at time step t. 4. y_t is the output at time step t. 5. f and g are activation functions for the hidden and output layers, respectively. 6. W_{hh}, W_{xh}, W_{hy} are weight matrices for the hidden-to-hidden, input-to-hidden, and hidden-to-output connections, respectively. 7. b_h and b_y are bias terms for the hidden and output layers, respectively.

$$y_t = g(W_{hy}h_t + b_y) \tag{2.5}$$

The Long Short-Term Memory (LSTM) recurrent neural network has become a ubiquitous method in sequential problems (e.g., language modelling, time series forecasting). This is so because it allows long-term dependencies to propagate through the network with the help of the control gates *input* and *forget*, which reduce the effect of the vanishing gradient issue in the vanilla RNN[8].

A more simple variant of the LSTM is the Gated Recurrent Unit (GRU) which combines the *input* and *forget* gates into an *update* gate for a model with fewer parameters and faster training [CCG21]. Nevertheless, due to the sequential nature of the LSTM, the training process cannot be parallelised across GPUs, i.e., the learning cannot be made quicker by more computational resources.

As noted by [GMH13], a limitation of the unidirectional RNNs is that they only make use of previous context in the sequence. To alleviate this and establish more complex relationships between words [SP97], propose a bi-directional architecture consisting of a forward and a backward RNN.

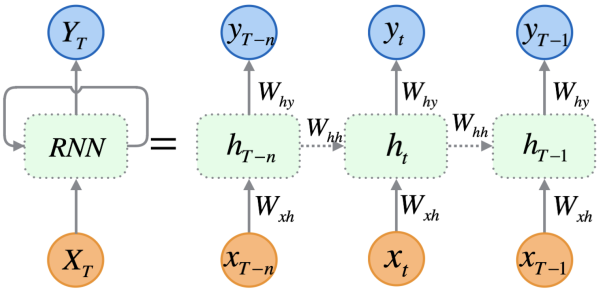
We can therefore represent the hidden layers (h_i) per time-step (where T is the sequence size) with the following notation: 1. $(\overrightarrow{h_1}, \ldots, \overrightarrow{h_T})$ are the forward hidden states from left to right (i.e., $x_1, \ldots, x_T$) and 2. $(\overleftarrow{h_1}, \ldots, \overleftarrow{h_T})$ are the backward hidden states from right to left (i.e., $x_T, \ldots, x_1$) (Figure 2.2b).

Then, for a single word x_i, its respective *annotation* (i.e., condensed representation) is constructed by the concatenation of the forward and backward hidden states - $h_i = [\overrightarrow{h_i^T}; \overleftarrow{h_i^T}]^T$ as specified in [BCB16].

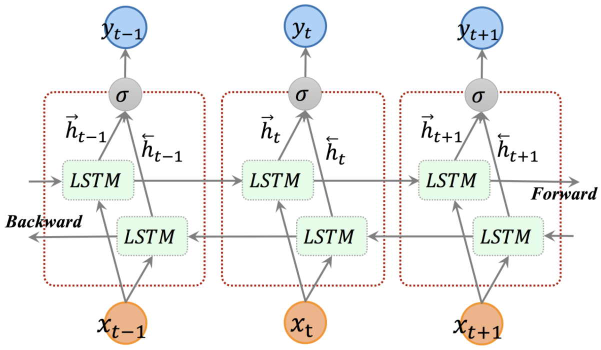
2.6 Encoder-Decoder and Attention

However, to deal with many-to-many sequence-to-sequence problems (e.g., machine translation, speech recognition, abstractive text summarisation) a new type of neural architecture is necessary [SVL14]. and [CvMG$^+$14] introduced the Encoder-Decoder network (Figure 2.3) with 1. the encoder being an RNN that maps an input sequence $(x_1, x_2, \ldots, x_n)$ to a continuous fixed-length

[8]We direct readers to [Bay15] where the authors demonstrate that the LSTM's "temporal" gradient is unaffected by the fixed weight factor W of the vanilla RNN that is driving the derivative to zero. This is ensured by the additional architecture unit - the *forget* gate, which learns to control the gradient flow in the network.



(a) Unfolded RNN



(b) Unfolded bi-directional LSTM

Figure 2.2: Unfolded recurrent architectures, taken from [CKW18]

context vector c and 2. the decoder, also an RNN, taking this vector and producing an output sequence $(y_1, y_2, \ldots, y_m)$. They train the two RNNs jointly, maximising the conditional probability of the target sequence given a source sequence, i.e., $p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T)$. For the selection of the neural model the authors had naturally chosen LSTM and GRU, respectively, due to the resolution of the vanishing gradient problem (as discussed in Section 2.5) and for the time being [SVL14] managed to achieve state-of-the-art results in machine translation.

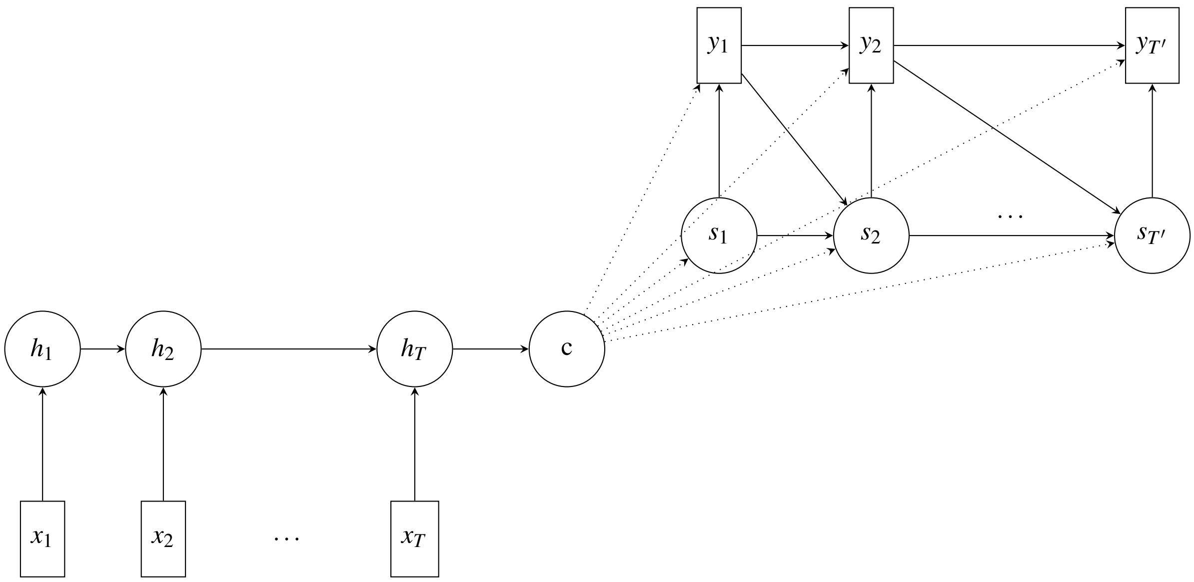


Figure 2.3: Encoder-Decoder Schema

It is in [BCB16] where the authors suppose that the fixed-length vector results in a bottleneck such that the longer the sequences, the worse the compression performance is for neural networks. Meanwhile, they propose to replace the fixed-length vector with a variable-length one which searches for the most relevant information from the source sequence. This is achieved through an alignment model which accepts as input the produced annotations from the encoder $h_1, \ldots, h_T$ (see Section 2.5 for details on how these are generated). Then the context vector c_i which is now distinct for each input word (unlike in [CvMG$^+$14, SVL14]) becomes a weighted sum[9] of the annotations h_j (Eq.2.6) and each weight α_{ij} is calculated as the normalized attention score (Eq.2.7). The alignment model is itself a neural network trained jointly with the encoder-decoder system and it evaluates the importance e_{ij} of an annotation h_j in generating a new state s_i and output token y_i (Eq.2.8). Intuitively, as noted in [GLT20], this attention mechanism computes a weight distribution on the input sequence and *attends* (i.e., assigns a larger weight) to the most relevant parts.

$$c_i = \sum_j^T \alpha_{ij} h_j \tag{2.6}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^T \exp(e_{ik})} \tag{2.7}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.8}$$

2.7 Transformers

The Transformer [VSP$^+$17] is another sequence-to-sequence architecture which follows the overall encoder-decoder architecture [SVL14], but differs in the following aspects:

1. *Internal architecture* - instead of RNNs [CvMG$^+$14], the encoder and the decoder have multiple identical transformer blocks based on fully-connected feed-forward neural networks with a newly introduced concept of self-attention (Figure 2.5). Although, prior to that, recurrent hidden layers have been used for compressing sequences into a context vector, the authors of the transformer extend the attention from [BCB16] to generate salient context-aware representations of the input sequence.

[9] And hence it is known as the *additive attention*.

2. *Self-Attention and Global context* - while RNN architectures struggle to deal with long-range dependencies (due to the vanishing gradient problem and the last-layer bottleneck [BCB16]), the authors propose *causal*[10] self-attention to allow capturing of global context. This mechanism has two key components (Fig.2.4):

   - *The scaled dot-product* (Eq.2.9) computes the attention weights (see Section 2.6) to be used for generating a weighted representation of the input sequence. Each token from the input sequence is represented as a q-query, k-key, and v-value vector (packed together into linearly-projected[11] Q, K, V matrices, respectively). Meanwhile, the dot product computes the similarity score between Q - the current token's focus and K - the context of the other tokens, whereas the scaling factor $\sqrt{d_k}$ prevents against extreme differences in softmax calculation - leading to slow convergence. Once the attention weights have been calculated, they are multiplied with V for a new context-aware representation combining information from other tokens in the sequence. This mechanism allows the Transformer to learn relationships between any two tokens and hence the significant computational load $O(n^2)$. Nevertheless, since these computations can be performed independently for each token, the entire input sequence can be processed simultaneously unlike in RNN architectures.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.9}$$

   - *The multi-head attention* [VSP$^+$17] is a mechanism that allows the Transformer to learn multiple representations of the input sequence. This is achieved by stacking multiple attention heads (each with its own Q, K, V matrices) and concatenating their outputs (Figure 2.4). For each head h_i, the Q, K, and V vectors will be linearly projected with different weight matrices W_i^Q, W_i^K, and W_i^V, respectively. The intuition is that each attention head learns different aspects of the relationships that exist among inputs (e.g., syntactic, semantic, and discourse relationships [JM00]) and the concatenation of the different representations allows the Transformer to learn a richer overall representation

[10] Causal attention is a special case of self-attention where the attention weights are computed only from the past tokens. This is achieved by masking the future tokens in the attention score computation (Figure 2.6a). We direct readers to [JM00] for a detailed discussion on Transformer language modelling.

[11] The query vectors, key vectors, and value vectors are linearly projected from the input token embeddings using separate learnable weight matrices.

of the input sequence.

3. *Positional Encoding* - the Transformer does not have a recurrent hidden layer to capture the sequential nature of the input sequence. Instead, the authors [VSP+17] propose adding a positional encoding to the input embeddings as a way to capture positional information.

4. *Training Details* - Because of the non-sequential nature of the Transformer, the heavy self-attention computations can be parallelised on modern hardware - GPUs and TPUs which makes this model easy during training and inference time. Additionally, the proposed dot-product attention is practically much faster and more space-efficient in comparison to the additive one [BCB16], due to highly optimized matrix multiplication libraries. Nevertheless, at least two computational drawbacks are that Transformers require an additional positional embedding and also have a quadratic complexity when calculating the self-attention which is extremely expensive for long sequences [JM00].

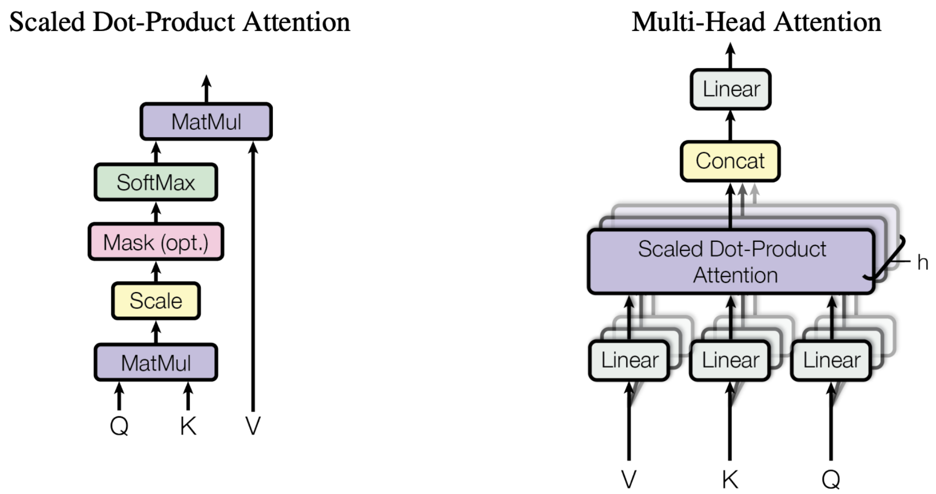


Figure 2.4: Scaled dot-product and multi-head attention, taken from [VSP+17]

2.8 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a Transformer-based language model that was proposed by Google AI in 2018 [DCLT19]. It revolutionised the field of NLP

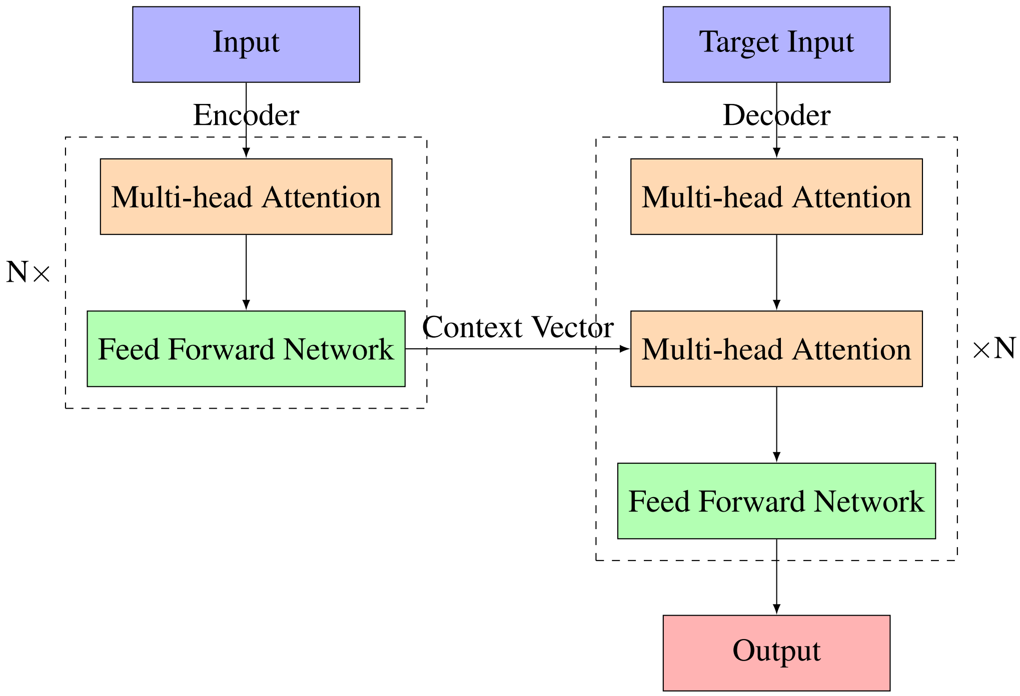


Figure 2.5: Simplified Transformer encoder-decoder architecture

by introducing a new pre-training paradigm that outperformed previous state-of-the-art models on a wide range of NLP tasks while being easily applicable to autoregressive generation problems (e.g., abstractive summarization and machine translation). We will discuss the main components of BERT and how it differs from the original Transformer model.

1. *Bidirectional Nature* - [JM00] argue that the uni-directional nature of the Transformer (i.e., causal self-attention) is a drawback that prevents it from capturing the full context of a sentence, especially when applied to sequence classification and labelling tasks. BERT addresses this issue by using a bidirectional Transformer architecture that allows the model to capture the full context of a sentence. This contextualisation is achieved by allowing the self-attention mechanism to range over the entire input as visible from the query-key comparisons in Figure 2.6b.

2. *Pre-training* - BERT is a model pre-trained on two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP).

   (a) *Masked Language Model* - BERT uses a pre-training task called Masked Language

q_1k_1	$-\infty$	$-\infty$	$-\infty$
q_2k_1	q_2k_2	$-\infty$	$-\infty$
q_3k_1	q_3k_2	q_3k_3	$-\infty$
q_4k_1	q_4k_2	q_4k_3	q_4k_4

(a) Transformer

q_1k_1	q_1k_2	q_1k_3	q_1k_4
q_2k_1	q_2k_2	q_2k_3	q_2k_4
q_3k_1	q_3k_2	q_3k_3	q_3k_4
q_4k_1	q_4k_2	q_4k_3	q_4k_4

(b) BERT

Figure 2.6: Comparison of query-key dot product representations for Transformer and BERT models.

Model (MLM) to learn the representations of words in a sentence. The authors [DCLT19] propose masking 15% of the input tokens at random and then training the model to predict the masked tokens. To make it more accessible for fine-tuning where the [MASK] token is not available, they replace it with a random or with the original token with probabilities of 10% each.

(b) *Next Sentence Prediction* - BERT is also pre-trained on Next Sentence Prediction (NSP) to learn the representations of sentences in a document. To understand the relationship between two sentences, BERT uses a binary classification task where the model is trained to predict whether a sentence B is following another sentence A. The authors [DCLT19] capture the sentence structure with the help of two new tokens - [CLS] (added before the sentence pair) and [SEP] (inserted in-between sentences), which are essential for the fine-tuning process (Figure 2.7). They further propose using 50% of the training data as positive examples (i.e., A and B are consecutive sentences) and 50% as negative examples (i.e., B is a random sentence from the corpus).

3. *Fine-tuning for Sequence Classification* - Unlike in Word2Vec [MCCD13] where the word embeddings are *static* (i.e., related to the single word token only), BERT learns *contextualised word embeddings* which can produce different representations for the same word depending on the context around it. Here, instead of using the output of the last hidden layer (as we do with RNNs and the Transformer), the authors [DCLT19] propose a *sentence embedding* y_{CLS} that summarizes the entire sequence of hidden states - the output vector of the model for the [CLS] token. The reasoning is that BERT is pre-trained with the [CLS] token being prepended to the input sequence (during the NSP task) and it can be used as an

aggregate representation of the entire sequence. Therefore, fine-tuning for text classification amounts to learning the probability distribution over the possible labels (e.g., 0/1 for a binary task) from the linearly-projected y_{CLS}, i.e. $\mathrm{softmax}(y_{CLS}W_C + b)$, complying with the typical supervised learning paradigm. A slight difference, as [JM00] suggest, is that the backpropagation can affect not only the classifier but also the pretrained language model (resulting in minimal changes in practice).

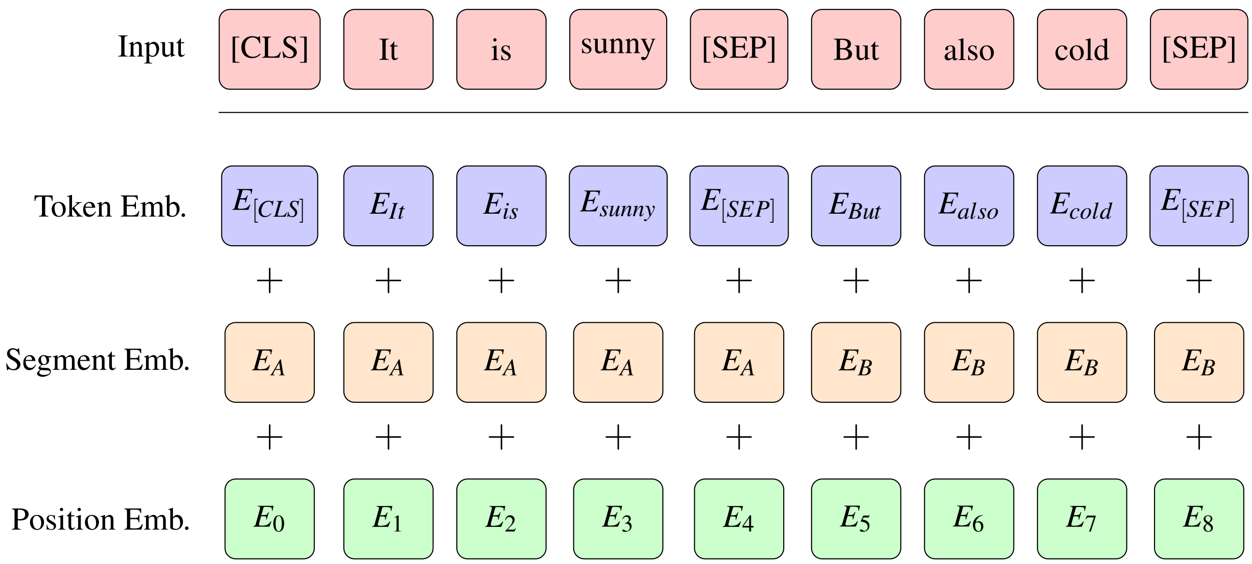


Figure 2.7: BERT: Input Embeddings

2.9 FinBERT

FinBERT[12] [YUH20] is a pre-trained domain-specific language model based on BERT [DCLT19] (Section 2.8) that is trained on a total of $4.9B$ tokens from financial corpora: 1. Corporate Reports 10-K & 10-Q (introduced in Section 1.1), 2. Earnings Call Transcripts (discussing financial performance, business updates, and future expectations), and 3. Analyst Reports (providing an in-depth textual analysis of the company and an earnings forecast). This makes FinBERT a natural choice for our experiments, as the language of the training data is very similar to the one of the UK annual reports (Section 1.2). Furthermore, the authors demonstrate that FinBERT outperforms

[12] `https://github.com/yya518/FinBERT`

BERT on three financial sentiment classification tasks, which is why we select to fine-tune it on our extractive summarisation task.

2.10 Text Summarisation

Text summarisation is the task of transforming a piece of text into a shorter version that retains the most important information. There are two overarching categories: extractive and abstractive text summarisation. The former formulates the problem as a subset selection problem by returning only the most salient text excerpts from the original document [ZLC$^+$20], while the latter aims to generate content anew, similar to how humans would do.

We will outline some key models from the FNS21 (the previous year competition) that inspired our work below:

- **Extractive SentenceBERT [GSL21])**: The authors employ an unsupervised summariser based on K-Means clustering of sentences encoded with SentenceBERT [RG19]. However, their embeddings are pre-trained on general text, and they suggest that employing in-domain language models would result in a better performance.

- **AMUSE** [LV21]: The authors design an ETS system comprised of the following steps 1. shortening of report with an existing Genetic Algorithm [LL13], 2. encoding sentences with BERT vectors, and 3. performing binary classification with LSTMs for salient sentence extraction. They suggest that further work should incorporate (a) efficient preliminary sentence removal, and (b) additional neural modelling stages for the representation and detection of relevant input text parts.

- **Hybrid model with RL** [ZSEHR21]: The authors train a joint extractive-abstractive summarisation model with reinforcement learning optimised for the ROUGE-2 F1 metric. Their networks are based on attentive LSTMs augmented with an additional copy mechanism [VFJ15] achieving the second highest F1 score in the FNS21 competition.

- **T5-LONG-EXTRACT** [Orz21]: The author used T5 [RSR$^+$20a] for an extractive model fine–tuned to generate the beginning of an abstractive summary and find the closest match of the output in the report's full text. This is the best performing algorithm in the FNS21 competition but also the first to consider transformer models from an extractive-abstractive perspective in the FNP workshops so far. It serves as an inspiration for the best English FNS22 model [EHZR+22] and other contestants [KGMB22, FRM+22].

Additionally, we briefly describe the FNS22 competitors' approaches, against which we perform our evaluation in Section 4.2:

- **T5 [PC22]**: This model is entirely based on the FNS21's T5-LONG-EXTRACT approach [Orz21] and achieves the best ROUGE-2 $F1$ score of 0.374 on the English testing set. Unfortunately, results on the validation set are not made available.

- **Top-K Narrative Extractor [SVK+22]**: The authors implement a narrative classification method recognising unique report sections, their titles (see Section 1.2), and the Table of Content. They further propose an algorithm for word allocation to assemble various sentences from different sections into a single summary. To fit in the $1,000$ word limit, they use a Top-k summarizer, which extracts the first k words from a given text. Their system achieves a ROUGE-2 $F1$ score of 0.425 on the English validation set, while ranking in the top three overall for Spanish and Greek report summarisation.

- **Longformer-Encoder-Decoder (LED) [KGMB22]**: The authors use a general transformer-based encoder-decoder model building on the LongFormer [BPC20] for an efficient calculation of long-range attention. They fine-tune the model on the FNS21 dataset to generate in an abstractive manner the start of summarising paragraphs, which are then extracted from the original text similaraly to [Orz21]. While their LED model performs well in recognising the beginning of the summary, it struggles with identifying the end, and hence their summarisation suffers from a lower ROUGE-2 score of 0.302 on the English validation set.

- **mT5** [FRM+22]: The authors propose the usage of a multilingual pre-trained text-to-text transformer [RSR+20b] to generate the start of a report's summary in the same way as [Orz21, KGMB22]. This system is ranked best overall for the three languages, while achieving a ROUGE-2 $F1$ score of 0.365 on the English validation set. The authors also explore unsupervised methods based on the multilingual BERT model [DCLT19] generating representations for blocks (i.e., spans) of 64 tokens for each summary and report. They further cluster the summary blocks with k-means and compute the cosine similarity between them and the report blocks. While this unsupervised method does not perform best for English, it proves to be very effective for Spanish and Greek, where the number of reports is significantly smaller than the one in English.

In this work we will be solely exploring the extractive summarisation method, and more specifically - the *supervised neural-based* (i.e., RNN, Transformer) type and the *unsupervised graph-based* (i.e., TextRank, LexRank) type.

2.11 TextRank

TextRank [MT04] is a graph-based unsupervised algorithm for extractive summarisation with the following key components:

1. **Sentence similarity**: TextRank computes the similarity between two sentences based on either the *cosine similarity* [13] of their vectors like bag-of-words, Word2Vec, FastText (see Eq.2.10 and Example 1) or the *Jaccard similarity* [14] of their sets of words (see Eq.2.11 and Example 1),

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.10}$$

EXAMPLE 2.11.1 **Calculating Cosine Similarity**

If vector A is $[1,1,0]$ and vector B is $[1,0,1]$, we can calculate their cosine similarity as follows:
Firstly, we calculate the dot product of the two vectors:

$$A \cdot B = (1x1) + (1x0) + (0x1) = 1$$

Next, we compute the magnitude (Euclidean norm) of each vector:

$$\|\mathbf{A}\| = \sqrt{1^2 + 1^2 + 0^2} = \sqrt{2}$$

and

$$\|\mathbf{B}\| = \sqrt{1^2 + 0^2 + 1^2} = \sqrt{2}$$

[13] Cosine similarity is defined as the dot product of two vectors divided by the product of their norms.
[14] Jaccard similarity is defined as the size of the intersection divided by the size of the union of two sets.

Finally, we calculate the cosine similarity:

$$\cos(\theta) = \cos(\mathbf{A}, \mathbf{B}) = \frac{1}{\sqrt{2}\sqrt{2}} = \frac{1}{2}$$

$$J(A,B) = \frac{|A \cup B|}{|A \cap B|} \tag{2.11}$$

EXAMPLE 2.11.2 **Calculating Jaccard Similarity**

If sentence A is "I love apples" and sentence B is "I love oranges", then:

- the word intersection between A and B is {"I", "love"}
- while the word union is {"I", "love", "apples", "oranges"}.

Therefore,

$$J(A,B) = \frac{2}{4} = \frac{1}{2},$$

which indicates that A and B share 50% of their unique words.

2. **Graph Construction**: TextRank represents sentences as nodes in a graph, and the similarity between each two sentences is represented as an edge between them.

3. **Sentence ranking**: The constructed graph is passed into the PageRank algorithm [BP98] that assigns each sentence a score based on the importance of its neighbours. The final summary is then assembled from the selection of the top k sentences with the highest scores.

Nevertheless, [SGWM20] report two considerable weaknesses of TextRank (hence the need for a better baseline - LexRank):

1. **Extraneous words** - TextRank does not penalise the extraneous words (i.e., words that do not add any essential information to the sentence) which can artificially increase the PageRank score (i.e., the importance) of a sentence.

2. **Frequent words** - There is no weighting applied regarding the frequency (or rarity) of words in the sentence, which can lead to a bias towards sentences with more frequent words.

As a countermeasure, *stop words* (e.g, “a”, “the”, “and”, etc) can be removed and *tf-idf* can be integrated into the similarity metric for a more balanced scoring mechanism. LexRank [ER04] builds up and resolves the issues of this algorithm, and we will discuss it in the next section.

2.12 LexRank

LexRank [ER04] is another unsupervised extractive summarisation method consistently used as a baseline in the FNS challenges over the years. It retrieves the most salient document sentences by computing their importance based on *eigenvector centrality*. To do that the algorithm creates a graph where each sentence represents a node and each edge is a weight between two nodes [SGWM20]. The sentences are encoded as bag-of-words vectors of size N - the vocabulary size, and the weight metric is a combination of tf-idf (Eq.2.3) and cosine similarity - Eq.2.12.

$$\text{tf\_idf\_cosine\_similarity}(s_1, s_2) = \frac{\sum_{t \in T} \text{tf-idf}(t, s_1, D) \cdot \text{tf-idf}(t, s_2, D)}{\sqrt{\sum_{t \in T} \text{tf-idf}(t, s_1, D)^2} \cdot \sqrt{\sum_{t \in T} \text{tf-idf}(t, s_2, D)^2}} \tag{2.12}$$

where t is a term, d is a document within a collection of documents/sentences D.

Also, s_1 and s_2 are two sentences and T represents the set of all terms in both of them while $tf(t,d)$ denotes the term frequency of t in d, and $idf(t,D)$ is the inverse document frequency of t in the collection D.

The authors further propose finding the most important sentences by 1. applying a threshold for the creation of edges with Eq.2.12, 2. building an adjacency matrix and normalizing it to produce *transition probabilities*, 3. computing in an iterative fashion the *eigenvector centrality* until convergence, and finally 4. ranking sentences based on their *lexical* PageRank [BP98] score similarly to TextRank [MT04].

2.13 Evaluation Metrics

In this section we discuss the evaluation metrics used in our project, specifically - the performance metrics derived from the confusion matrix (Section 2.13.1), and the ROUGE metrics (Section 2.13.2).

2.13.1 Confusion Matrix

The confusion matrix is an essential tool to visualise and help assessing the performance of trained classifiers against the true labels y_i.

For the problem of binary classification, it is a square matrix (Table 2.1) that displays the following key elements:

- True Positives (TP): Correct predictions of the positive class.
- True Negatives (TN): Correct predictions of the negative class.
- False Positives (FP): Incorrect predictions of the positive class (Type I error).
- False Negatives (FN): Incorrect predictions of the negative class (Type II error).

where for the problem of extractive text summarisation, the positive and negative classes correspond to *summary* and *non-summary* sentences, respectively. These matrix elements can be further combined into informative classification metrics:

- Accuracy: Proportion of correctly classified instances out of the total instances. Formulated as $\frac{TP+TN}{TP+TN+FP+FN}$.
- Precision (or Positive Predictive Value): Proportion of true positive instances out of all instances predicted as positive. Formulated as $\frac{TP}{TP+FP}$.
- Recall (or Sensitivity): Proportion of true positive instances out of all actual positive instances. Formulated as $\frac{TP}{TP+FN}$.
- F1-score: Harmonic mean of precision and recall (i.e., the trade-off between the two). Formulated as $2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$.

2.13.2 ROUGE

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [Lin04] is a popular recall-based evaluation metric for summarisation tasks which measures the overlapping units (i.e., n-grams) of a predicted summary c and a reference summary c^*. Here we briefly describe some of the most common variants of ROUGE (especially in the context of the FNS Task):

	Actual	
	Positive	**Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Table 2.1: Confusion Matrix

1. **ROUGE-N** measures the overlap of n-grams (sequences of n words) between c and c^*. For example, the ROUGE-1 and ROUGE-2 correspond to the overlap measurement of unigrams and bigrams, respectively, calculated as:

$$ROUGE-N = \frac{\sum_{S \in R} \sum_{n-gram \in S} count_{match}(n-gram)}{\sum_{S \in R} \sum_{n-gram \in S} count(n-gram)} \tag{2.13}$$

We further provide a demonstration on how to compute the ROUGE-2 $F1$ score which is the official FNS metric (see Example 1)

EXAMPLE 2.13.1 **Calculating ROUGE-2 $F1$ score**

We briefly demonstrate how to compute ROUGE-2 for a sentence pair c = "Fox can run" and c^* = "Fox can walk".
Here, the bigrams of c are $\{(\text{“Fox”}, \text{“can”}), (\text{“can”}, \text{“run”})\}$.
Meanwhile, the bigrams of c^* are $\{(\text{“Fox”}, \text{“can”}), (\text{“can”}, \text{“walk”})\}$.
The intersection of the two bigram sets is $\{(\text{“Fox”}, \text{“can”})\}$, which is the only bigram that appears in both c and c^*.
Therefore, to compute the recall of ROUGE-2:

$$\text{ROUGE-2 recall} = \frac{\text{Overlapping bigrams}}{\text{Total bigrams in reference summary}} = \frac{1}{2} = 0.5$$

Next, we can find the precision of ROUGE-2:

$$\text{ROUGE-2 precision} = \frac{\text{Overlapping bigrams}}{\text{Total bigrams in candidate summary}} = \frac{1}{2} = 0.5$$

Finally, we can calculate the F1 score of ROUGE-2:

$$\text{ROUGE-2 F1 score} = \frac{2 \times \text{ROUGE-2 recall} \times \text{ROUGE-2 precision}}{\text{ROUGE-2 recall} + \text{ROUGE-2 precision}} = \frac{2 \times 0.5 \times 0.5}{0.5 + 0.5} = 0.5$$

2. **ROUGE-L** uses the longest common subsequence (LCS) to measure the similarity between c and c^*. The LCS is the longest sequence of words that appears in both summaries *in the same order*, but *not necessarily consecutively*. Thus, ROUGE-L is less sensitive to word order compared to ROUGE-N and can capture longer-range dependencies.

While ROUGE scores are explainable and easily implementable, they have certain limitations like 1. inability to capture semantic similarity [ABK22], 2. insensitivity to summary coherence [CMSE13], and 3. lack of correlation with human evaluation [LL10]. Despite these limitations, ROUGE remains a widely-used evaluation measure, particularly in extractive summarization and we use it in this work as well.

Methodology

In this chapter we describe the FNS text reports and the issues they present (Section 3.1). We specify how we pre-process the data, and formulate our summarisation problem as a sentence extraction task (Section 3.2). We further outline how we deal with the data imbalance problem (Section 3.3) and introduce the two different architectures we use for the summarisation task (Sections 3.4, and 3.5). Finally, we describe the training (Section 3.6) and the hyper-parameter tuning process (Section 3.7).

3.1 Data

The data for the FNS22 task is a collection of narrative parts of annual reports, converted from PDF to plain text. As discussed in Section 1.4 due to the rich visual representations in the PDFs, the resulting text suffers from various problems like (a) *spacing inconsistencies* - mixing of tab-space word delimiters, over-segmentation (i.e., a split into incoherent chunks) and under-segmentation (i.e., merging of unrelated words), (b) *symbol encoding issues* - introduction of unreadable non-alphanumeric characters, and (c) *formatting issues* - words having different casing, hyphenation at the end of a line, etc (d) *conversion of tables to text* - financial figures spanning over multiple lines and being mixed with the text. (Figure 3.1).

To address these issues, we have developed a rigorous data cleaning pipeline that achieves the following key objectives: (a) handles space-tab mixing via hand-crafted rules (derived from observation[1]), (b) retains alphanumeric characters, punctuation, spaces, financial symbols, and (c) removes sentences shorter than 3 words.

[1]E.g., for some of the lines, characters were separated by spaces and words with tabs, hence the need for a custom rule.

1.
```
 Following my appointment as Chief
E x e c u t i v e in J u l y 2 0 1 0 ,g r e ate r
e m p h a s i s
h a s b e e n p l a c e d o n f u l fi l l in g t h e
s u p p l y o f
tonnage due under legacy contracts and
```

2.
```
However,   the   Directors   further   believe   that   additional
capital    could    be    deployed    to  beneficial    effect.
```

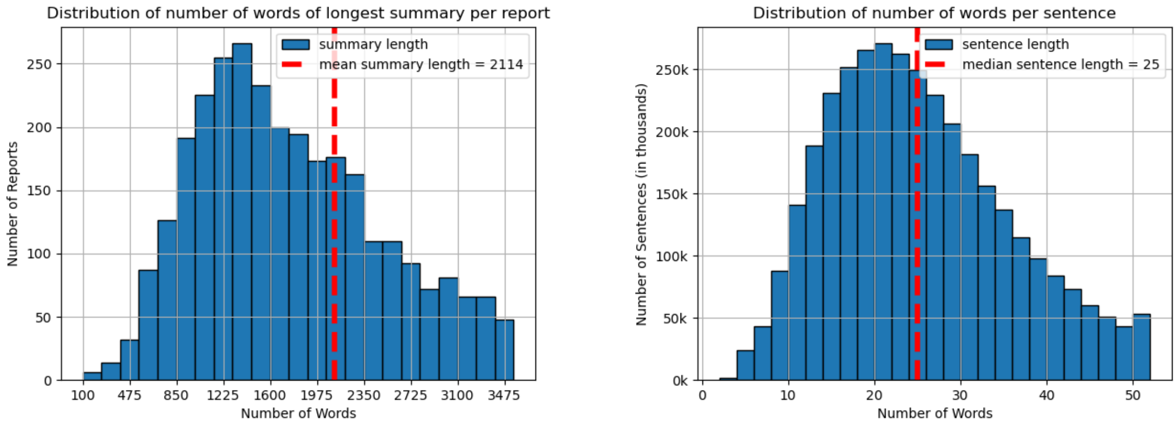
3.
```
Opening net book amount 116,635 35,624 166,754 319,013
Additions 51,380 7,647 307,546 366,573
```

4.
```
  This means that buyers can
_0_@uk_ar06_front.indd   5 20/04/2007   09:13:30 05
@UK PLC
Annual Report and Accounts 2006
use our network to purchase from their suppliers.
```

Figure 3.1: PDF-to-text conversion issues.

As discussed in Section 1.4, the annual reports are extremely long documents with an average length reported at 80 pages [LV21]. Each one has at least two–three gold summaries provided by the FNS22 organisers, and we compute some statistics (a) helpful for grasping the nature of the output text, but also (b) useful for the evaluation of the summarisation models. On one hand, we can see that the average number of words in the longest summary is over $2,000$ (Figure 3.2a), while the FNS22 regulations specify an expected output of at most $1,000$ words. Furthermore, as we are not competing in the FNS22 task, for simplicity, during evaluation we will generate only summaries with at most 40 sentences (capped at $1,000$ words). We arrive at this number by observing that the median number of words in the longest summaries is 25 (Figure 3.2b), and calculating that $\frac{1{,}000\text{words}}{25\text{words}} = 40$ sentences can suffice.

As we were only provided with the training and the validation FNS22 datasets (Table 1.1),



(a) Number of words in longest report summary (b) Number of words in training sentences

Figure 3.2: Distribution of number of words in training sentences and report summaries

we decided to treat the validation set as a testing set (Table 3.1)[2] and perform our own training-validation data split on a sentence level instead due to the significant variation in report lengths [LV21]. Specifically, we use a 90–10 *stratified split* (i.e., the label distribution is retained in both sets) for

Data Type	Training + Validation	Testing	Total
Report full text	2,998	363	3,361

Table 3.1: Training-Validation-Testing Data Split

training and validation, respectively (similar to [SPPŠ+22]). We are aware that a validation and a training sentence can come from the same report, but we claim that this is not problematic for the following reasons:

1. Sentences are *de-contextualised* (i.e., without references or dependencies to others, taken out of context) and *shuffled*.
2. Sentences contain a *great deal of textual noise* due to the PDF-to-text conversion.
3. Annual reports are *numerous* but also *extremely long* (i.e., containing a lot of sentences).

[2] We observed that two of the annual report files were empty, hence the difference of 3,361 and 3,363 (Table 1.1 without testing set).

Therefore, we believe that the training and validation sentences are to a large extent independent, and as for the process of sentence extraction, we refer you to Section 3.2 for an in-depth discussion.

3.2 Summarisation Problem Formulation

We approach the annual report summarisation problem from a supervised perspective - we cast the task of Extractive Text Summarisation (ETS) as a binary classification problem defined on the sentence level. More formally, we can describe the annual report as $d = \{s_1, s_2, \dots, s_n\}$, where d is a document, represented in terms of sentences $s_i,\ 1 \leq i \leq n$ [Liu19].

Then, a candidate summary[3] can be $c = \{s_1, s_2, \dots, s_k | s_i \in d\},\ 0 \leq k \leq n.$ We further need to define the *gold summary*, c^* for a document d. In the case of the FNS22 task, there are at least two summaries per report, hence we will use the following notation for the set of all gold summaries for each document $C^* = \{c_1^*, c_2^*, \dots, c_p^*\}$. Furthermore, the supervised learning labels are $y_i \in \{1, 0\}$ for each sentence s_i in d if the sentence is or is not in ***any*** of the gold summaries c_j^* for that document. We argue that in order to increase the positive samples (i.e., the summarizing sentences) we should not restrict ourselves to just one gold summary in the training process unlike [Orz21]. Our goal is to achieve better latent feature extraction of summaries through the employment of all existing data, hence using ***any*** gold summaries. However, we are aware that this approach is more likely to encounter standard ETS issues, specifically - extracted summary sentences could be retrieved from unrelated paragraphs in the report. This can cause the "dangling anaphora" phenomenon, i.e. de-contextualised extracts are stitched together and can mislead the reader due to out-of-context references [Lin09].

While some authors [ZSEHR21] follow the greedy ROUGE-maximisation method of matching summary sentences to document sentences (established in [NZZ17]), we approach the problem in a more practical and faster fashion. After manual observation of the reports against their gold summaries, it became clear that almost for all sentences belonging to c_i^*, there was an exact match with a sentence in the whole annual report d.

This hypothesis was proven correct by one of the FNS contestants [Orz21] who reported that 99.4% of the summaries were included in the report as whole subsequences. Hence, after having pre-processed the text documents we iteratively match the sentences and generate the binary classification labels ($\{1, 0\}$ representing *summary* and *non-summary*, respectively) for both the training

[3] A candidate summary is generated from a model m_i but it is not yet a *best summary*.

and testing datasets.

We perform the sentence extraction as discussed above to produce a dataset (training and validation combined) of $3,554,800$ and $361,703$ sentences for classes 0 and 1, respectively. Additionally, the training and validation sets are split in a 90–10 stratified fashion (similarly to [SPPŠ+22]).

3.3 Under-sampling and Data Augmentation

Due to the fact that the annual reports are extremely long while the summaries are very short, the sentence dataset is highly imbalanced with a ratio of around 1:10. Therefore, we took two[4] different approaches to balance the classes:

1. **Random under-sampling** – As described in [Wei13, WHB23], we *randomly remove 90% of the sentences* from the majority class (i.e., non-summary sentences) to produce a ratio of 1:1 summary to non-summary sentences (Figure 3.2).

	initial data	**90% under-sampled training**	**validation**
label 0	3,199,319	319,931	355,481
label 1	325,533	325,533	36,170

Table 3.2: 90% Random Under-sampling

2. **Data Augmentation** – We use the *back-translation* technique [HKHC18] to generate new sentences from the minority class (i.e., summary sentences), followed by a 80% random under-sampling of the majority class (Figure 3.3). For that purpose, we translate all training summary sentences from English to French and back to English with the help of the MarianMT model [JDGD+18][5]. The resulting dataset is then directly injected during the training process. Refer to Section 3.7 for the experiments with the different data balancing techniques.

[4] We also tried a third approach that is to augment sentences with the help of DINO [SS21] used for high-quality semantic augmentation, but we did not manage to recreate the desired output.

[5] https://huggingface.co/Helsinki-NLP/opus-mt-fr-en

	initial data	**80% under-sampled training**	**augmented training**	**validation**
label 0	3,199,319	639,863	639,863	355,481
label 1	325,533	325,533	651,066	36,170

Table 3.3: Data Augmentation + 80% Random Under-sampling

3.4 Recurrent Extractor Model

As our main recurrent model we propose a GRU-based architecture [CvMG$^+$14], inspired by [ZSEHR21] and depicted in Figure 3.3.

The model consists of a word embedding layer, a fully-connected feed-forward neural network (FCFFNN), two bidirectional gated recurrent units (GRU) layers, a dot-product attention layer, and a linear projection layer with softmax activation.

The word embedding layer is used to convert the pre-processed input sentence into a vector representation. One of our implementation innovations is that we use FinText's FastText word embeddings [RZP21] because they (a) are character-based and thus can handle noisy or out-of-vocabulary words, and (b) are pre-trained on large corpora of financial news, achieving considerable in-domain performance improvements over general-purpose embeddings.

We use the FCFFNN layer to *map the vectorized sentences to a higher-level representation* (similar to [SDEB20]) capturing more complex features or patterns from the input text, but also to *reduce the dimensionality* of the input.

Two stacked GRU layers are used to *extract the latent recurrent features* from the compressed vector representation in both directions - forward and backward (refer to Section 2.5 for details).

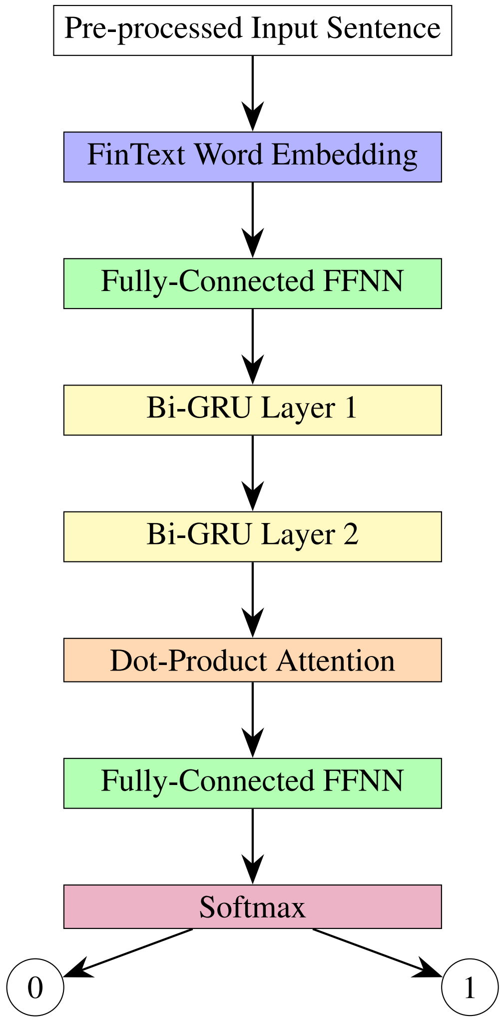


Figure 3.3: GRU-based extractive model

We further implement the scaled dot-product attention (Eq.2.7 from [VSP$^+$17]) to compute a new weighted context-aware representation from the extracted features by the GRU layers. The final layer is a fully-connected feed-forward neural network (FCFFNN) with a softmax activation function, which is used to *map the latent features to a binary classification* of the input sentence.

3.5 Fine-tuning FinBERT

Financial BERT (FinBERT) [YUH20] is a transformer-based language model pre-trained on financial communication (Section 2.9). We propose to fine-tune it for extractive summarisation on the sentence-level FNS datasets (Section 3.2), utilizing its demonstrated financial sentiment classification strengths. For this purpose, we tokenize the input sentences using the FinBERT tokenizer and feed them to the model as a sequence of tokens with a maximum length of 128 per sentence. The input tokens are then encoded as a combined representation of the token, segment, and position embeddings, which allow the model to attend to the overall semantic and syntactic aspects of the input sentence. As discussed in Section 2.8, by prepending a special CLS token this allows the model to condense the whole sequence representation into a single vector. This vector can now be used as an input to a classification head [JM00], where FinBERT's authors specify it as a Linear layer with a softmax activation function. Similarly to a typical supervised learning task (Section 2.1), the model is fine-tuned to predict the binary label of the input sentence (i.e., whether it is a summary sentence or not). We further summarise the fine-tuning process in Figure 3.4 and we specify the hyperparameters we used in Section 3.6.

3.6 Training

We trained both models on Tesla V100-SXM2-16GB[6] and provide the following specifications:

1. **RNN architectures** - below we provide the common general details, however for an in-depth discussion on the hyperparameter tuning, see Section 3.7.

    - **Loss function:** Binary cross-entropy loss
    - **Optimizer:** Adam [KB17]

[6]We extend our gratitude to the University of Manchester's Computational Shared Facility (CSF) for kindly agreeing to provide us with the computational resources for this research.

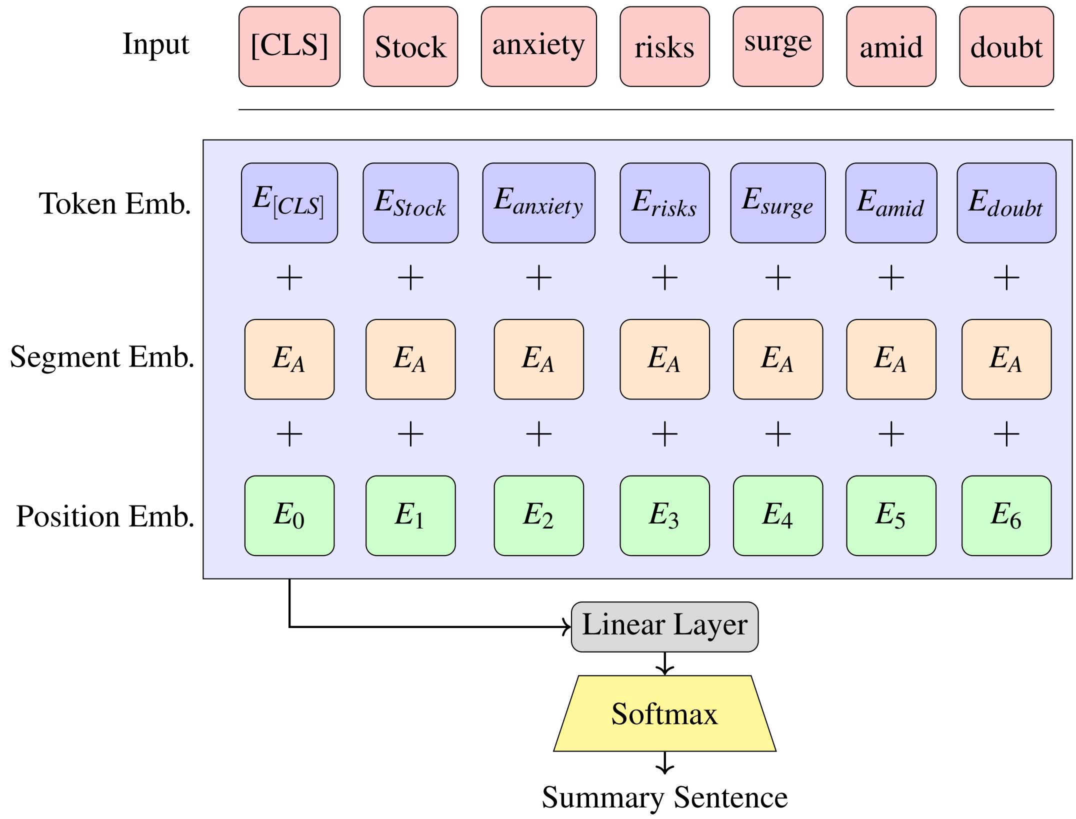


Figure 3.4: Fine-tuning FinBERT for extractive summarisation / sentence classification

- **Batch size:** 32
- **Epochs:** set to 60, but due to early stopping, the model practically trains for less than 10 epochs
- **Early stopping:** patience (i.e., the number of epochs to wait for improvement based on validation loss) set to 1

2. **FinBERT** - We followed the prescribed fine-tuning specifications provided in the original BERT paper [DCLT19]:

   - **Loss function:** Binary cross-entropy loss
   - **Optimizer:** Adam [KB17]
   - **Batch size:** 32

- **Epochs:** 3 - we found that the model started to over-fit after 3 epochs
- **Learning rate:** 2e-5
- **Weight decay:** 0.01

3.7 Recurrent Extractor: Hyperparameter Tuning

We have experimented with a number of hyperparameters for our recurrent model, including the use of a FCFFNN, the recurrence type, the hidden units, the effect of applying attention, the dropout rate, and the effect of data augmentation.

For the analysis we will be extensively using the test accuracy, $F1$-score, and the *summary recall* metric. The latter is defined as the ratio of the number of correctly predicted summary sentences to the total number of summary sentences in the test dataset. We consider this metric to be extremely relevant because in the context of extractive summarisation, our goal is to minimise the Type II error (i.e., the number of sentences that should be in the summary but are not). Our reasoning is that our classifier must be as good as possible in recognising salient sentences (i.e., summarising sentences) even if it introduces some false positives (i.e., non-summarising sentences). In practice, the user can always remove irrelevant sentences, but it is much harder to add sentences that should have already been in the summary.

We would also like to remind the reader of the label distribution per dataset (Tables 3.2 and 3.3). It is worth noting that although we have managed to balance the training set with the help of random under-sampling and data augmentation, our validation and testing sets are left unchanged (i.e., they are imbalanced).

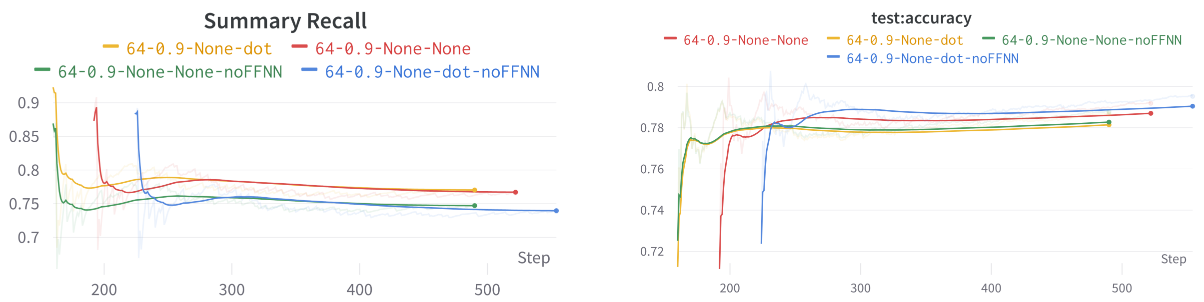
Throughout the hyperparameter tuning, we keep the learning rate fixed to 0.001[7] and employ the following naming convention for our models: `A-B-C-D-F`, where:

- `A` - the number of hidden units
- `B` - the under-sampling percentage (e.g., `0.9` means 90% of the majority class is removed)
- `C` - the data augmentation strategy (i.e., `None` or `fr` - French back-translation)
- `D` - the type of attention (i.e., `None` or `dot`-product attention [VSP+17])

[7]Preliminary results showed that considerably different values such as 0.1, 0.01, 0.0005, lead to unsatisfiable results.

- `F` - another hyperparameter (e.g., `noFFNN` - no feed-forward neural network , or `dropout rate` of 0.25)

Each sentence in the report is represented as a (100, 300)–sized word embeddings vector, where 100 is the longest possible sentence length (i.e., implying long sentences are trimmed) and 300 is the dimensionality of the word embeddings. We test the effect of inserting an FCFFNN layer between the word embeddings and the GRU layers (each with 64 hidden units) and arrive at the following results: adding an FCFFNN layer increases Summary Recall by 2.5% (Fig. 3.5a), but marginally reduces Test Accuracy by less than 1% (Fig. 3.5b). We attribute the increase in Summary Recall to the fact that the FCFFNN layer is able to extract an additional mix of features from the word embeddings, which are then used by the GRU layers to make better predictions. As for the Test Accuracy, we believe that the small decrease is insignificant and we therefore choose to use the FCFFNN layer in our final model.



(a) Effect of FCFFNN layer on summary recall

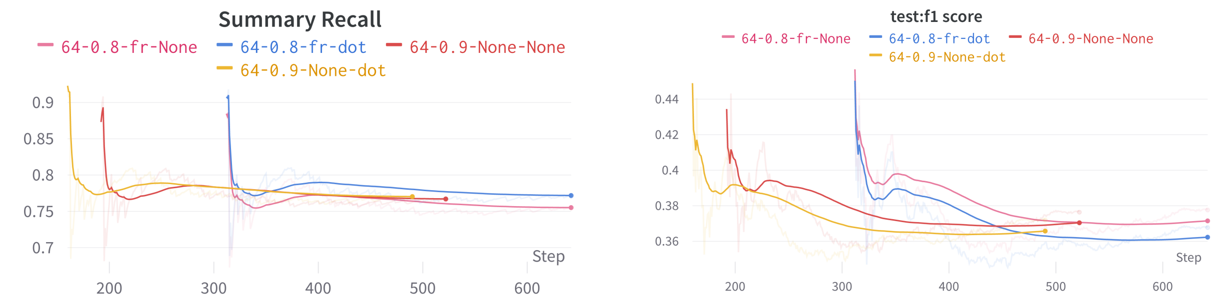
(b) Effect of FCFFNN layer on test accuracy

Figure 3.5: Effect of FCFFNN layer on summary recall and test accuracy

We also explore the effect of using back-translated data (Section 3.3) on the model performance. Results shown in Fig. 3.6 suggest that the data augmentation does not improve the $F1$-score or the summary recall with a statistically significant margin. At the same time, it seems to be amplifying the effect of the scaled dot-product attention (Section 2.6, Figure 3.6a). While we are disappointed by these results, we believe that there can be a number of reasons for this. For a machine translated sentence s_i^m and its original sentence s_i, we *hypothesise* that:

1. s_i^m contain a similar amount of noise as s_i (due to the pdf-to-text conversion process);

2. s_i^m does not introduce enough variation to s_i (i.e., s_i^m and s_i are too similar);

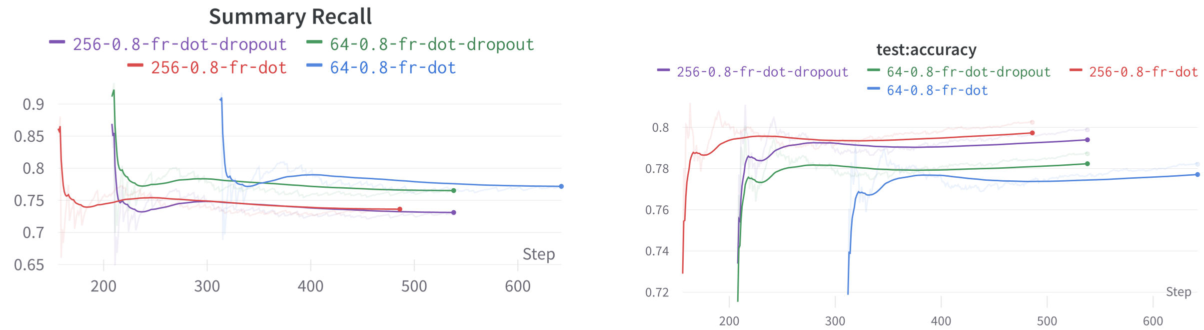


(a) Effect of data augmentation on summary recall (b) Effect of data augmentation on $F1$-score

Figure 3.6: Effect of back-translation data augmentation on summary recall and $F1$-score

3. while financial language itself is very domain-specific, it is not very semantically diverse (i.e., metaphors, idioms, etc. are limited in use);

In our proposed architecture we choose a bidirectional GRU (Bi-GRU) instead of a Bi-LSTM because (a) GRUs have a simpler structure than LSTMs and are easier to train (Section 2.5), and (b) our experiments show that GRUs outperform LSTMs with 1% in terms of Summary Recall. Furthermore, in terms of the hidden units, we select 64 over 256 because: (a) the architecture has $143,938$ and $2,050,306$ parameters, respectively (i.e., 256 would result in an over-parameterised model), (b) almost 4% increase in Summary Recall (Fig. 3.7a), although a 2% decrease in Test Accuracy (Fig. 3.7b), if we use 64 hidden units.



(a) Summary Recall: 256 vs 64 hidden units + dropout (b) Test Accuracy: 256 vs 64 hidden units + dropout

Figure 3.7: Effect of hidden units and dropout on summary recall and test accuracy

From the above experiments it is hard to make any certain conclusions on the effect of a dropout of 0.25[8] or the use of an attention mechanism (Section 2.6). We believe that only one *single-head attention* is not sufficient to learn all the complex relationships between words in the sentences. Our reasoning is that because a particular head specializes to only specific language aspects (i.e., syntactic, semantic, etc) [CKLM19]), for future experiments it would be much more reasonable to use multiple heads instead. Nevertheless, there still are some practical benefits of attention to our extractive summarisation system which we will discuss in Chapter 4.

Overall, four variations of the GRU model engage our focus based on the results above and our design decisions (Figure 3.6a). For ease of comparison during evaluation (Section 4.2) we will refer to their architectures as FinRNN (Financial RNN):

1. FinRNN-base with 90% under–sampling

2. FinRNN-base with 90% under–sampling + attention

3. FinRNN-base with 80% under–sampling + data augmentation

4. FinRNN-base with 80% under–sampling + data augmentation + attention

Because our binary classification metrics do not demonstrate any significant differences between them, we will test their performance with ROUGE during the evaluation phase (Chapter 4).

3.8 Summary Generation

After having extracted all reports' sentences into appropriate datasets with binary labels (Section 3.2), and having trained our models (i.e., FinRNN and FinBERT) on them (Section 3.6), we can now predict the summarising probability (i.e., potential) of each sentence in a report (Figure 3.8). Nevertheless, to produce a summary, we must decide which sentences to include as the FNS Task imposes a word limit of *at most* $1,000$. After computing overall textual statistics from the annual report (Figures 3.2 and 3.2a), we found that 40 sentences is a reasonable upper bound for the number of sentences in a summary (Section 3.1). Therefore, we decided to use the top 40 sentences (capped at $1,000$ words) with the highest summarising probability as the summary. To ensure maximal coherence, we sort the sentences in natural order (i.e., sort by their position

[8] It would make more sense for its application in the over-parameterised model, though this does not seem to be the case (Fig. 3.7).

in the report). While we are aware that this has the potential of causing the "dangling anaphora" phenomenon (Section 3.2), our aim in this project is to outperform the baseline methods (i.e., TextRank and LexRank) on the ROUGE metrics (Section 2.13.2).

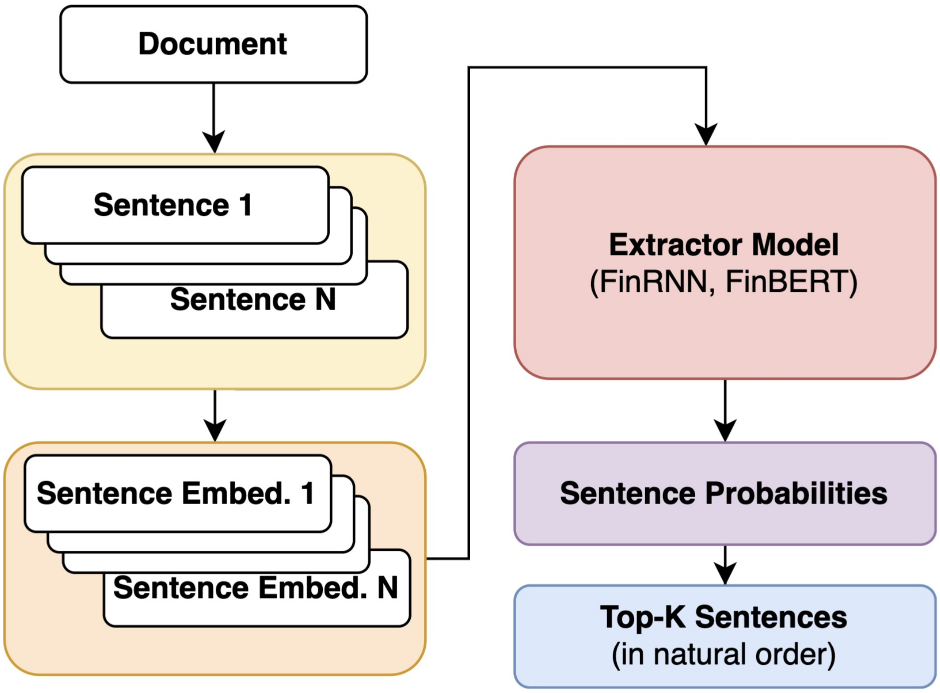


Figure 3.8: Extractive Summarisation Pipeline

Evaluation

In this chapter we outline the evaluation process of our models. For that purpose, we describe how we use the ROUGE metrics (Section 2.13.2) to compute the similarity between the gold and candidate summaries, and how we arrive at the final evaluation aggregation (Section 4.1). Afterwards, we provide the results of our evaluation (Section 4.2), where we compare our models to the baselines and official FNS models. Finally, we discuss the qualitative limitations of our produced summaries (Section 4.3) before we conclude our work in Chapter 5.

4.1 Evaluation Mechanism

Now we will provide a formal formulation of the summary generation process (Section 3.8) - to assemble a candidate summary c_i we prepare the sentences s^i_j from a report d_m (e.g., embed or tokenize sentences for the FinRNN and FinBERT, respectively). Afterwards, we feed them into our model computing the output summary probabilities p^i_j, and then we select the top-k sentences ($k = 40$, Section 3.1) based on the highest sentence probabilities p^i_j. Finally, we concatenate them to form the summary c_i in natural order (i.e., the order in the input text), but also trim it to the maximum length of $1,000$ words (Section 1.4).

In general, to assess the quality of a candidate summary c, we measure its similarity with the gold summary c^* based on their subsequence (i.e., n-gram) overlap $R = (c, c^*)$, where R is the ROUGE-2[1] metric [Lin04]. For the FNS task due to the extractive nature of our approach we will

[1] We use a slightly different but faster version of ROUGE compared to the official metric [Lin04]. It can be accessed at: `https://github.com/pltrdy/rouge`.
The FNS evaluation metric is the $F1$-score of ROUGE-2, and we will use it for the final evaluation.

evaluate our models based on the ROUGE-L-maximising[2] $c_i^{*\max}$ gold summary (Section 2.13.2), i.e.,

$$c^{*\max} = \underset{c^* \in C^*}{\text{argmax}} \text{ROUGE-L}(c, c_i^*) \tag{4.1}$$

where C^* is the set of gold summaries for a given report d, and $c_i^{*\max}$ is the gold summary with maximal ROUGE-L score with the candidate summary c.

The intuition is that by extracting multiple sentences from the report, our generated candidate summary can retain sentences from *any* of the gold summaries. Hence, there must be at least one such gold summary where the longest common subsequence overlap (ROUGE-L) is maximal. The practical implications are that two models, m_1 and m_2 can produce two different candidate summaries c_1 and c_2, respectively. Their individual evaluation is based on gold summaries c_1^* and c_2^* (which can be the same when the candidates c_1 and c_2 are identical). This guarantees that we are always comparing candidate summaries based on their maximal evaluation scores (i.e., their maximal summarising potential).

Therefore, the final evaluation score for a model m is the average ROUGE-2 score[3] between the candidate summaries c_i and their corresponding *ROUGE-L-maximising gold summaries* $c_i^{*\max}$, i.e.,

$$r_m = \frac{1}{|C|} \sum_{i=1}^{|C|} \text{ROUGE-2}(c_i, c_i^{*\max}) \tag{4.2}$$

where $|C|$ is the number of candidate summaries c_i.

4.2 Quantitative Evaluation

Following this evaluation mechanism, we compare our models in terms of their ROUGE metrics against the baselines and official FNS models. For that purpose, we produce two tables, namely:

- Table 4.1, which compares **all** of our models against the baselines;

[2]We say ROUGE-L-maximising for conciseness, because ROUGE is a metric with precision and recall components, which are combined into a single $F1$ score. It is precisely the $F1$ score that we maximise. However, we are aware this introduces more complexity to our naming convention, hence we will use the term ROUGE-L-maximising.

[3]Here, once again we mean the $F1$-measure of ROUGE-2.

- Table 4.2, which compares our **best** models against the FNS ones. Results are on the official validation set (used as a testing set for our models);

We again wish to remind the reader that we were not provided with the FNS22 testing set (Section 3.1).

In our FNS comparison we include the following models, namely the:

1. T5 model [EHZR+22] (testing data is only available), which is the completely based on the T5-LONG-EXTRACT and also the best English model in that edition;

2. mT5 model [FRM+22], which is the best model overall for all languages in the competition (multilingual for English, Spanish, and Greek);

3. Longformer-Encoder-Decoder (LED) [KGMB22]

4. Top-K Narrative Extractor (NE) [SVK+22] ranking in the top three models overall for Spanish and Greek;

We have the official English validation results for all but the T5 [EHZR+22], for which we are going to use their testing evaluations (we are aware that this comparison is somewhat unfair).

Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
TextRank [MT04]	0.220	0.064	0.196
LexRank [ER04]	0.250	0.086	0.227
FinRNN-base + attention	0.221	0.062	0.204
FinRNN-base	0.220	0.063	0.201
FinRNN-base + back-translation	0.266	0.100	0.247
FinRNN-base + attention + back-translation	0.276	0.106	0.249
FinBERT-base + back-translation	0.490	0.321	0.468
FinBERT-base	**0.544**	**0.382**	**0.524**

Table 4.1: ROUGE $F1$ Scores of our FinRNN and FinBERT models against baselines TextRank and LexRank

We can provide the following commentary on the results:

- Our best performing model, FinBERT-base, which is a pre-trained on financial communication documents [YUH20], achieves an average ROUGE-2 score of 0.382, which outperforms

Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
TextRank [MT04]	0.220	0.064	0.196
LexRank [ER04]	0.250	0.086	0.227
mT5 [FRM+22]	0.440	0.301	0.423
LED [KGMB22]	0.442	0.302	0.434
T5 [EHZR+22] (*official testing set*)	0.496	0.374	0.487
Top-K NE [SVK+22]	**0.546**	**0.425**	-
FinRNN-base + attention + back-translation (ours)	*0.276*	*0.106*	*0.249*
FinBERT-base (ours)	*0.544*	*0.382*	*0.524*

Table 4.2: ROUGE $F1$ Scores on official FNS22 validation set (used as testing in our models)

by 0.081 on the validation set the best performing model **overall** in the FNS22 competition - *mT5* [FRM+22]. Furthermore, our FinBERT-base also seems to *slightly outperform* the **best English model** in the FNS22 - T5 [EHZR+22] with 0.008, though this comparison is not entirely fair as we are not using the same datasets. Regarding the other FNS models, we observe that our FinBERT-base beats the LED [KGMB22] with a similar margin as the mT5 [FRM+22]. However, the Top-K NE [SVK+22] remains with the highest ROUGE-1 and ROUGE-2 $F1$ measures. Surprisingly, data augmentation does not improve the performance of our FinBERT-base (Table 4.1), and we believe this was caused by the back-translation hypothesis we made in Section 3.7. Nevertheless, both models clearly outperform the official baselines: LexRank [ER04] and TextRank [MT04] (Table 4.1).

- The FinRNN-base + attention + back-translation model is the best performing model out of all recurrent neural architectures. While preliminary binary classification results did not show any considerable differences between the models, clearly (a) the attention mechanism helps the model to better recognise the summarising sentences (i.e., attends to the most descriptive linguistic features), and (b) the back-translation data augmentation significantly improves the practical performance of the model (i.e., the probability distribution of the summarising sentences), which is clearly not the case for our FinBERT model. Additionally, we must note that except of the Top-K NE [SVK+22], all other FNS models are transformer-based, hence they have more complex architectures and attention mechanisms than our FinRNN-based models with its single-head attention (Sections 2.7 and 3.4).

- At the same time, we acknowledge that the universal summarisation baselines: LexRank and TextRank, outperform our simple FinRNN models (Table 4.1), and we attribute this to both:
    1. the lack of sufficient descriptive training data from the positive class (i.e., the summarising sentences, Table 3.2);
    2. the 90% random under-sampling of the majority class data (see Section 3.1);
    3. the bias towards long and complex sentences (Section 4.3);
    4. the summary generation process of selecting the top-k sentences based on their sorted probabilities (Sections 4.3 and 5.2);

4.3 Qualitative Discussion

After having established the quantitative performance of our models, we now turn to the qualitative discussion of the results. For a random annual report we generated a summary using the FinBERT-base + data augmentation model and the FinRNN-base + attention model (see Figures 6.3, 6.4). where green colour indicates the summarising sentences, and red colour – the non-summarising sentences). We chose these models over the best ones because they have slightly lower ROUGE scores, and make more mistakes, which will help us identify the issues in the summarisation process. We can make the following conclusions based on observation:

1. The FinBERT model has *around 50% of its contents* belonging to any of the gold summaries (Figure 6.3), while all other sentences look very convincing in terms of their summarising potential (i.e., they are informative of the financial situation but also concise)
2. At the same time, the FinRNN has the opposite characteristics: (a) none of its sentences are in the gold summary (Figure 6.4), while (b) all of them are very long, containing uninformative but diverse sets of words, which in turn results in higher ROUGE scores. This clearly represents the problem of using ROUGE as a metric for summarisation since it is only measures the lexical overlap, while being semantically unaware [ABK22].

While we acknowledge that the FinRNN model seems to be *biased towards long and noisy sentences*, we must note that in the example only 5 sentences have been generated to fit the word limit. Therefore, we believe the summary generation process (i.e., the mechanism to combine predicted sentences into a single summary) further exacerbates the accuracy of the summarisation.

Although, the practical results from Figures 6.4 might seem disappointing, we must once again remind the reader that the reports are extremely long with an average number of sentences and words at around $2,700$ and $58,000$, respectively [LV21]. Meanwhile, we are constrained to producing a summary of at most 40 sentences (Section 3.1) or $1,000$ words (Section 1.4).

Conclusion

In this chapter we summarise the main contributions of our work, highlighting its innovational aspects and discussing its limitations.

5.1 Summary of Achievements

In this project we have explored the problem of summarising UK annual reports. To deal with the significant amount of noise in the plain text of these glossy documents, we have built a rigorous pre-processing pipeline (Section 3.1). We have further implemented a sentence extraction phase where we generate binary labels (1 being summary, 0 - non-summary) from the reports and their multiple gold summaries (Section 3.2). Once our datasets are created, we (a) design a Recurrent Neural Network (RNN) architecture (Section 3.4), and (b) fine-tune a financial Transformer model - FinBERT (Section 2.9), training them in a supervised manner for a binary classification task (Sections 3.6 and 3.7). We quantitatively evaluate our models with the ROUGE metric and demonstrate them outperforming traditional baselines (Chapter 4).

Furthermore, at least for FinBERT we observe a clear ROUGE-2 improvement on the validation set over the best overall FNS22 model [FRM+22]. Additionally, we show that our FinBERT also achieves competitive performance with the best FNS22 English model, although noting that the our models are tested on different data (Section 3.1). We also discuss the quality of the produced summaries (Section 4.3), and in Section 5.2 we describe in more depth the limitations of our system and possible solutions.

In terms of *innovational aspects* of our project in the context of the FNS challenge, we are the first to our knowledge that:

- *Integrate FinText word embeddings* - While some FNS21 competitors use general-domain sentence embeddings based on BERT [LV21, GSL21], we represent sentences as a vector of word embeddings purpose-built for financial text analysis [RZP21].

- *Perform back-translation as data augmentation* - In contrast to approaches where only the first 10% of the annual report is used [Orz21], we over-sample the summarising sentences (minority class) by back-translating from French.

- *Fine-tune FinBERT [YUH20] for extractive summarisation* - Transformer-based models have become increasingly popular in the FNP22 Workshop [KGMB22, PC22], where some have used FinBERT for classifying definitions [GSNS22], detecting hypernyms [PCHH22], and classifying financial sentiments [SPPŠ+22]. However, we are the first to adapt FinBERT for extractive summarisation.

5.2 Discussion of Limitations

Although, both our models have outperformed the baselines on ROUGE-2 and the fine-tuned FinBERT has achieved competitive performance for the FNS22 task, there are several limitations that we would like to discuss:

- *Sentence embedding* - Although, we note our use of domain-specific FastText word embeddings due to their ability to handle noise and outperform general-domain embeddings [RZP21], we do not perform any sentence–level aggregation (i.e., dimensionality reduction) like averaging to condense the overall representation. While, this was a deliberate design decision to better capture the relationship between individual words, our sentence vectors became of size $(100, 300)$ instead of $(300,)$, which became more computationally expensive. Although, we are aware that FastText [BGJM17] provides average-pooling for any sequence, we were pessimistic of using it due to the loss of word order information (e.g., "the company is good" being represented just as "is the company good"). Therefore, a limitation of our work is that we do not investigate the impact of using sentence embeddings (be it with positional encoding or average-pooling) on the performance of our models.

- *Non-exhaustive evaluation* - Due to the FNS models being proprietary, and also evaluated on the official testing set, we are unable to make a more comprehensive comparison with the other models. However, in an ideal scenario we would perform further quantitative and qualitative evaluation of our performance. We would also like to investigate more in-depth the effect of under-sampling and the random generator on the classification capabilities of our models and compare with simply using the first 10% of the annual reports as in [Orz21].

- *Summary Generation* - In our work we take top k sentences based on the model's output probability distribution (Chapter 4). However, this is a very simplistic approach to summarisation that (a) introduces incoherence issues (like the *dangling anaphora phenomenon* from Section 3.2), (b) trims the last sentence to fit the $1,000$ word limit, and it (c) does not account for the *informativeness* of the individual sentences. To address the incoherence issues, we can try resolving the coreferences in the either through a graph-based approach on the generated summary [SK16], or by introducing a more complex encoder architecture that represents and attends to entities as well as sentences [HK21]. Regarding the trimming heuristic, a natural improvement can be to use text compression techniques for the final predicted sentence [GHI22,KM02]. As for the third point, we believe this is very exacerbating reason why our FinRNN architecture returns a summary without any single whole–sentence overlap with the gold standard (Fig. 6.4). Instead, what [ZSEHR21] propose is a reinforcement learning approach which incorporates the *sentence-level* ROUGE-2 score with the whole gold summary. While this method is much more sophisticated, it conveys the intuitive idea that the top-k sentences comprising the *optimal candidate summary* should be *greedily maximising the global summary-level ROUGE-2* score.

5.3 Future Work

While in our project we only consider the narrative summarisation of financial reports already converted to plain text, we propose the following pipeline as a direction for future work:

1. *PDF-to-Text* - Integrate into the summarisation system a PDF-to-Text conversion tool for annual reports like the CFIE-FRSE[1] [EHRY$^+$19], which also extracts the text into 8 generic section headers (Section 1.2).

2. *Text-to-Summary* - Implement an extractive method that addresses the limitations from Section 5.2, or alternatively, an abstractive method producing *lay summaries* for non-expert

[1] `https://github.com/drelhaj/CFIE-FRSE`

users [VJM+23, GQWC20].

3. *Text-to-Analysis* - Apply NLP techniques like sentiment analysis [Ara19], named entity recognition [ZZ22], and detection of forward-looking sentences [ŠPŽ21], to extract useful information from the summary and the text. Additionally, important financial disclosure characteristics as amount, tone, and transparency [L+10, Li11] would be beneficial for AF researchers and users.

4. *Packaged Software* - Build a drag-and-drop software application that allows users to upload a PDF file of an annual report, where the backend will perform the steps above and return a summary (see example Figures 6.5, and 6.6 in Appendix). The suggested textual analysis features could be integrated as interactive visual elements. Furthermore, through recognising company names, the system could also provide dashboard of news and stock prices with the help of company-to-identifier mapping [EHAR+19] (i.e., getting the company ticker, e.g., *NASDAQ: AAPL* for Apple Inc).

Bibliography

[ABK22] Mousumi Akter, Naman Bansal, and Shubhra Kanti Karmaker. Revisiting automatic evaluation of extractive summarization task: Can we do better than ROUGE? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1547–1560, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[Ara19] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019.

[Bay15] Justin Simon Bayer. Learning sequence representations, 2015.

[BCB16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[BGJM17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30:107–117, 1998.

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150, 2020.

[BSF94] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[CCG21] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. A comparison of LSTM and GRU networks for learning symbolic sequences. *CoRR*, abs/2107.02248, 2021.

[CHW19] Eddy Cardinaels, Stephan Hollander, and Brian White. Automatic summarization of earnings releases: attributes and effects on investors' judgments. *Review of Accounting Studies*, 24, 09 2019.

[CKLM19] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics.

[CKW18] Zhiyong Cui, Ruimin Ke, and Yinhai Wang. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. *CoRR*, abs/1801.02143, 2018.

[CMSE13] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. Towards coherent multi-document summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[CvMG+14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[EDH19] Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262, Florence, Italy, July 2019. Association for Computational Linguistics.

[EHAR+19] Mahmoud El-Haj, Paulo Alves, Paul Rayson, Martin Walker, and Steven Young. Retrieving, classifying and analysing narrative commentary in unstructured (glossy) annual reports published as pdf files. *Accounting and Business Research*, 2019. Forthcoming.

[EHO22] Mahmoud El-Haj and Andrew Ogden. Financial narrative summarisation using a hybrid TF-IDF and clustering summariser: AO-lancs system at FNS 2022. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 79–82, Marseille, France, June 2022. European Language Resources Association.

[EHRW+19] Mahmoud El-Haj, Paul Rayson, Martin Walker, Steven Young, and Vasiliki Simaki. In search of meaning: Lessons, resources and next steps for computational analysis of financial discourse. *Journal of Business Finance & Accounting*, 46(3-4):265–306, 2019.

[EHRY+19] Mahmoud El Haj, Paul Edward Rayson, Steven Eric Young, Paulo Alves, and Carlos Herrero Zorita. *Multilingual Financial Narrative Processing: Analysing Annual Reports in English, Spanish and Portuguese*. World Scientific Publishing, 2019.

[EHRZ22] Mahmoud El-Haj, Paul Rayson, and Nadhem Zmandar, editors. *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, Marseille, France, June 2022. European Language Resources Association.

[EHZR+22] Mahmoud El-Haj, Nadhem Zmandar, Paul Rayson, Ahmed AbuRa'ed, Marina Litvak, Nikiforos Pittaras, George Giannakopoulos, Aris Kosmopoulos, Blanca Carbajo-Coronado, and Antonio Moreno-Sandoval. The financial narrative summarisation shared task (FNS 2022). In *Proceedings of the 4th Financial Narrative*

Processing Workshop @LREC2022, pages 43–52, Marseille, France, June 2022. European Language Resources Association.

[Ell98] Robert K Elliott. Accounting in the 21st century. 1998.

[ER04] Gunes Erkan and Dragomir R. Radev. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 2004.

[FRM+22] Negar Foroutan, Angelika Romanou, Stéphane Massonnet, Rémi Lebret, and Karl Aberer. Multilingual text summarization on financial documents. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 53–58, Marseille, France, June 2022. European Language Resources Association.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GHI22] Demian Gholipour Ghalandari, Chris Hokamp, and Georgiana Ifrim. Efficient unsupervised sentence compression by fine-tuning transformers with reinforcement learning, 2022.

[GLT20] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. *IEEE transactions on neural networks and learning systems*, 32(10):4291–4308, 2020.

[GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

[GQWC20] Yue Guo, Weijian Qiu, Yizhong Wang, and Trevor A. Cohen. Automated lay language summarization of biomedical scientific reviews. *ArXiv*, abs/2012.12573, 2020.

[GSL21] Tuba Gokhan, Phillip Smith, and Mark Lee. Extractive financial narrative summarisation using SentenceBERT based clustering. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 94–98, Lancaster, United Kingdom, 15-16 September 2021. Association for Computational Linguistics.

[GSNS22] Sohom Ghosh, Shovon Sengupta, Sudip Naskar, and Sunny Kumar Singh. FinRAD: Financial readability assessment dataset - 13,000+ definitions of financial terms for measuring readability. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 1–9, Marseille, France, June 2022. European Language Resources Association.

[HK21] Yin Jou Huang and Sadao Kurohashi. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2021.

[HKHC18] Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[JDGD+18] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in c++, 2018.

[JM00] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition, 2000.

[Jon72] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[KB16] P Kriz and H Blomme. The future of corporate reporting–creating the dynamics for change. *International Federation of Accountants (IFAC), available at: www. ifac. org/global-knowledgegateway/viewpoints/future-corporate-reporting-creating-dynamics-change (accessed 29 May 2016)*, 2016.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[KGMB22] Urvashi Khanna, Samira Ghodratnama, Diego Molla, and Amin Beheshti. Transformer-based models for long document summarisation in financial domain. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 73–78, Marseille, France, June 2022. European Language Resources Association.

[KM02] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

[KVL21] Sophie Krimberg, Natalia Vanetik, and Marina Litvak. Summarization of financial documents with tf-idf weighting of multi-word terms. In *FNP*, 2021.

[L$^+$10] Feng Li et al. Textual analysis of corporate disclosures: A survey of the literature. *Journal of accounting literature*, 29(1):143–165, 2010.

[Li11] Feng Li. Textual analysis of corporate disclosures: A survey of the literature. 2011.

[Lin04] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[Lin09] Jimmy Lin. Summarization. In *Encyclopedia of Database Systems*, pages 2906–2910. Springer, Heidelberg, Germany, 2009.

[Liu19] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[LL10] Feifan Liu and Yang Liu. Exploring correlation between rouge and human evaluation on meeting summaries. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1):187–196, 2010.

[LL13] Marina Litvak and Mark Last. Multilingual single-document summarization with MUSE. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 77–81, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[LRU20] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 3 edition, 2020.

[Luh57] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1(4):309–317, oct 1957.

[LV21] Marina Litvak and Natalia Vanetik. Summarization of financial reports with AMUSE. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 31–36, Lancaster, United Kingdom, 15-16 September 2021. Association for Computational Linguistics.

[LY19] Craig Lewis and Steven Young. Fad or future? automated analysis of financial text and its implications for corporate reporting. *Accounting and Business Research*, 49(5):587–615, 2019.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

[MCCD13] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MT04] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[MYZ13] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[NZZ17] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[Ola15] Christopher Olah. Understanding lstm networks, 2015.

[Orz21] Mikhail Orzhenovskii. 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 2021. Association for Computational Linguistics.

[PC22] Manish Pant and Ankush Chopra. Multilingual financial documentation summarization by Team_Tredence for FNS2022. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 112–115, Marseille, France, June 2022. European Language Resources Association.

[PCHH22] Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, and Chu-Ren Huang. Discovering financial hypernyms by prompting masked language models. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 10–16, Marseille, France, June 2022. European Language Resources Association.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[RSR+20a] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 21(1), 2020.

[RSR+20b] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[RZP21] Eghbal Rahimikia, Stefan Zohren, and Ser-Huang Poon. Realised volatility forecasting: Machine learning via financial word embedding. *SSRN Electronic Journal*, preprint:1–20, July 2021.

[SDEB20] Tanik Saikh, Arkadipta De, Asif Ekbal, and Pushpak Bhattacharyya. A deep learning approach for automatic detection of fake news, 2020.

[SGWM20] Steven Shearing, Abigail S. Gertner, Ben Wellner, and Liz Merkhofer. Automated text summarization: A review and recommendations. 2020.

[SK16] Sheetal Sonawane and Parag Kulkarni. The role of coreference resolution in extractive summarization. In *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pages 351–356, 2016.

[SP97] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[SPPŠ+22] Timen Stepišnik-Perdih, Andraž Pelicon, Blaž Škrlj, Martin Žnidaršič, Igor Lončarski, and Senja Pollak. Sentiment classification by incorporating background knowledge from financial ontologies. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 17–26, Marseille, France, June 2022. European Language Resources Association.

[ŠPŽ21] Jan Štihec, Senja Pollak, and Martin Žnidaršič. Preliminary experimentation with combinations and extensions of forward-looking sentence detection wordlists. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 26–30, Lancaster, United Kingdom, 15-16 September 2021. Association for Computational Linguistics.

[SS21] Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models, 2021.

[SVK+22] Neelesh Shukla, Amit Vaid, Raghu Katikeri, Sangeeth Keeriyadath, and Msp Raja. DiMSum: Distributed and multilingual summarization of financial narratives. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 65–72, Marseille, France, June 2022. European Language Resources Association.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[SW11] Claude Sammut and Geoffrey Webb. *Encyclopedia of Machine Learning*. Springer, 2011.

[VFJ15] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes,

N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[VJM+23] Oliver Vinzelberg, Mark David Jenkins, Gordon Morison, David McMinn, and Zoe Tieges. Lay text summarisation using natural language processing: A narrative literature review, 2023.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[Wei13] Gary M. Weiss. *Foundations of Imbalanced Learning*, chapter 2, pages 13–41. John Wiley and Sons, Ltd, 2013.

[WHB23] Tarid Wongvorachan, Surina He, and Okan Bulut. A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining. *Information*, 14(1), 2023.

[XCWS13] Zhixiang Eddie Xu, Minmin Chen, Kilian Q. Weinberger, and Fei Sha. An alternative text representation to tf-idf and bag-of-words. *ArXiv*, abs/1301.6770, 2013.

[YUH20] Yi Yang, Mark Christopher Siy UY, and Allen Huang. Finbert: A pretrained language model for financial communications, 2020.

[ZEHR+21] Nadhem Zmandar, Mahmoud El-Haj, Paul Rayson, Ahmed Abura'Ed, Marina Litvak, Geroge Giannakopoulos, and Nikiforos Pittaras. The financial narrative summarisation shared task FNS 2021. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 120–125, Lancaster, United Kingdom, 15-16 September 2021. Association for Computational Linguistics.

[ZLC+20] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online, July 2020. Association for Computational Linguistics.

[ZSEHR21] Nadhem Zmandar, Abhishek Singh, Mahmoud El-Haj, and Paul Rayson. Joint abstractive and extractive method for long financial document summarization. In

Proceedings of the 3rd Financial Narrative Processing Workshop, pages 99–105, Lancaster, United Kingdom, 15-16 September 2021. Association for Computational Linguistics.

[ZZ22] Yuzhe Zhang and Hong Zhang. Finbert-mrc: financial named entity recognition using bert under the machine reading comprehension paradigm, 2022.

Appendix

6.1 UK annual report example

OXFAM Trustees' Annual Report 2021/22

A MESSAGE FROM OUR CHAIR

Charles Gurassa stands with local indigenous women leaders in Samburu county, Kenya, whilst meeting an organisation Oxfam partners with: 'Samburu Women Trust' - an indigenous women-led organisation that champion the rights of women and girls among indigenous communities. Photo: John Plastow

Thanks to our brilliant supporters we have been able to reach out to over 8 million people around the globe this year and support them in the fight against poverty and injustice. Thanks also to our supporters and the great work of our teams we have been able to end the year with a much-improved financial position and can move forward with confidence, and with an ambitious vision and plans for the future.

A huge personal thank you goes to all my colleagues, our staff and our thousands of wonderful volunteers for their tireless efforts this year, and continued pursuit of a world where everyone can thrive. Central to our vision is working with and through local partners embedded in their own communities. They are best placed to deliver the sustainable change that we all desire–their first-hand knowledge, skills and experience enables them to deliver broader and deeper impact to more people and more places than we could ever do on our own.

I had the privilege to witness this in Kenya earlier this year, where I saw first-hand our response to the growing hunger crisis in East Africa. I came back with greater appreciation of the enormous difficulties faced and how Oxfam and our local partners are in the frontline fighting the scourge of drought and supporting community groups and women in particular to access desperately needed resources. Standing in a barren field with three hundred women hearing how transformational it had been for them to directly receive much-needed cash payments from Oxfam in the early days of the food crisis was a genuinely humbling experience.

The crisis in East Africa has been, and remains, our highest priority – although it has not yet had the widespread attention it desperately merits. We estimate that every 36 seconds someone is likely to be dying of hunger in the region. Millions are facing the devastating effects of climate change they did not cause, alongside soaring food and fuel prices as a result of the war in Ukraine.

During the last 12 months, we responded to many other crises, including Ukraine. The public response to the crisis has been phenomenal, including through generous donations to the Disasters Emergency Committee appeal of which we are proud to be part. It demonstrates again the deep humanitarian instinct of so many

6

"A huge personal thank you goes to all my colleagues, our staff and our thousands of wonderful volunteers for their tireless efforts this year, and continued pursuit of a world where everyone can thrive."

MARY'S STORY

Oxfam partner the Community Technology Development Trust (CTDT) helped set up a community seed bank in Zambia in 2018. It was an initiative set up for safe keeping of seeds, some of which are in danger of becoming extinct. Farmer facilitators like Mary, who is featured on the next page, coordinate activities. They are trained to train other farmers.

The seed bank is a reserve for seeds in case of any form of calamity and also allows participating farmers to share seeds among themselves. About 500 farmers contribute to the seed bank.

The seed bank is not a place where farmers buy or sell seeds. The participating farmers are the ones that bring their seed to the seed bank to keep them safe. It's a reserve house for the seeds but also a place to learn about how to keep seeds and at the same time it's a place where farmers can share seeds.

With some of the plants/seeds, the reason they are growing those crops is because they are adaptable to the environments in relation to climate change. Other varieties of seeds are due to their preference of wanting to be able to sell crops.

Below: Eukeria gathers together with the participating farmers in the seed bank. Eukeria Samba is the District Field Officer for Community Technology Development Trust (CTDT). This photo was produced with funding by the European Union. Photo: Loliwe Phiri/Oxfam

Figure 6.1: Excerpts from Oxfam's 2021–2022 (glossy) annual report

Figure 6.2: Excerpts from Oxfam's 2021–2022 (glossy) annual report

6.2 Example Model Summaries

Paul, who is a chartered accountant and has a wealth of experience in senior finance positions, joins us as an independent non-executive director and chairman of the audit committee. The first point worth noting, and celebrating, is the strength of our financial performance. The visibility of revenues with a contracted order book of £5.7 billion, inflation-linked pricing, and the opportunity to increase utilisation of our infrastructure, places arqiva in a very strong position to continue to deliver stable profitable growth. In december 2013, the group established facilities in arqiva smart financing limited (a group company) that support the group's smart energy metering contracts by financing the purchase of communication hubs. We believe that the critical national infrastructure that we provide will continue to be in demand; people will continue to watch television, listen to radio, use mobile devices and consume increasing amounts of data. 3 main national commercial multiplexes refers to those considered to be most established. We thank clive for his valued counsel. With the timing and nature of britain's exit from the european union still uncertain, i would like to reiterate the sentiment from my statement in our 2016 annual report. The arqiva network, which forms part of that, is successfully transmitting and receiving test messages between dcc users (the energy companies) and consumer electricity and gas meters. He led the company through a period of growth and also a successful listing on the new y ork stock exchange. The trial is operational in london over the summer and showcases the solution, demonstrating what superfast connectivity will mean for uk business and residents. The strategy lays out the ambition and actions required by the government to create an environment that positions the uk as a global leader in the next generation of mobile technologies and digital communications. The board of directors regularly review health and safety reports in relation to the group's activities, employees and contractors. This is an exciting time for arqiva and i would like to thank all our employees across the business for their dedication and hard work, which has been central to our continued growth and success. We are now seeing the positive impact of these actions on our financial performance. It has been another excellent year for arqiva, as we continue to see the reward from the strategic capital investment decisions over recent years, with our operating cash flow after investing activities improving significantly. It is essential that businesses and consumers have access to seamless, uninterrupted communications and broadcast quality content anywhere and at any time. Be a great place to work by continuing to invest in our people build the group's knowledge and grow its expertise, led by a dynamic senior management team with a clear vision and proven track record. I am confident that our strategy, together with the support of our people, will continue to deliver our objectives and enable us to grow as a business. The triggering of article 50 to begin the uk's exit from the european union heightens the uncertainty over future policy and economic conditions. Excellent progress, reflected in our financial performance this financial year has been a year of significant progress. The achievement of this one million mark is a strong indication of the attractiveness of hybrid dtt / ip tv services in the uk where dtt remains the underlying delivery mechanism that has a core free-to-air linear content base with a variety of ott services on-top. Outlook our focused investment in the right project has resulted in the group benefitting from another year of impressive growth in revenue, earnings and cash generation. We will make further announcements as and when appropriate. 1 is a key measure of the group's financial performance. The strategy lays out the ambition and actions required to create an environment that positions the uk as a global leader in the next generation of mobile technologies and digital communications. The achievement of this one million mark is a strong indication of the attractiveness of hybrid dtt / ip tv services in the uk where dtt remains the underlying delivery mechanism that has a core freeto-air linear content base with a variety of ott services on-top. A pioneer in an always on, always connected world. What is particularly pleasing is that we have seen growth across each of our businesses. Shareholders strategic review earlier this year, arqiva's shareholders informed the board that they are jointly undertaking a strategic review of their investment, which may lead to a transaction involving their interests in arqiva. We have invested in the right activities that best utilise our assets and strengths, divested of noncore activities that lack strategic fit, and right sized the operational base of the business. This non-replicable asset base will support arqiva's leading position for the foreseeable future. These benefited from new hd channel sales, a new agreement with al jazeera media network for global teleport and distribution services and foreign exchange gains. This, combined with the refinancing exercise undertaken in the year, leaves us stronger as a business. Outlook this is an exciting time for the business with improvements in financial performance and operational efficiency. Arqiva has continued to meet its contractual milestones on its major contractual programmes, e.g. Arqiva has continued to achieve its target result for network availability' (see page 28). Good progress is being made to secure the next tranche of savings. Arqiva has invested substantially in infrastructure as a result of these contracts, which now result in recurring cash flows during the long-term operational phases of the networks. Hd services are seen as business critical, with big shows attracting the largest audience shares and therefore commanding the largest advertising revenues.

Figure 6.3: Example summary from the FinBERT-base + data augmentation model.

The risks and uncertainties referred to above include actions or decisions by governmental and regulatory bodies, or changes in the regulatory framework in which the group operates, which may impact the ability of the group to carry on its businesses; changes or advances in technology, and availability of resources such as spectrum, necessary to use new or existing technology, or customer and consumer preferences regarding technology; the performance of the markets in the uk, the eu and the wider region in which the group operates; the ability of the group to realise the benefits it expects from existing and future projects and investments it is undertaking or plans to or may undertake; the ability of the group to develop, expand and maintain its broadcast and telecommunications infrastructure; the ability of the group to obtain external financing or maintain sufficient capital to fund its existing and future investments and projects; the group's dependency on only a limited number of key customers for a large percentage of its revenue; and expectations as to revenues not under contract. Annual report and consolidated financial statements 2017 arqiva group limited (company reg 05254001) 93 11 other gains and losses notes year ended 30 june 2017 £m year ended 30 june 2016 £m foreign exchange on financing (8.3) (38.1) fair value (loss) / gain on derivative financial instruments 25 (104.2) 38.0 other gains and losses (112.5) (0.1) exceptional (loss) / profit on disposal of subsidiary 7,29 (5.2) 14.4 exceptional close out of swap arrangements 7 (15.4) - exceptional other gains and losses (20.6) 14.4 total other gains and losses (133.1) 14.3 foreign exchange on financing arises on the revaluation of the group's us dollar denominated debt (see note 23). 10 finance costs year ended 30 june 2017 £m year ended 30 june 2016 £m interest on bank overdrafts and loans 101.1 100.1 other loan interest 130.3 125.0 bank and other loan interest 231.4 225.1 amortisation of debt issue costs 13.1 12.2 interest on obligations under finance leases 1.0 1.0 shareholder loan note interest 316.6 278.5 other interest 17.0 19.4 total interest payable 579.1 536.2 less amounts included in the cost of qualifying assets - (1.7) unwinding of discount on provisions (see note 26) 3.0 1.8 total finance costs 582.1 536.3 the shareholder loan notes carry fixed interest rates of between 13.0% and 14.0%, payment of which can be deferred at the option of the group subject to certain conditions, qualification of which are subject to biannual review (see note 23). Arqiva group limited registered number 05254001 annual report for the year ended 30 june 2017 corporate information as at the date of this report (11 september 2017) group board of directors simon beresford-wylie (chief executive officer) mark braithwaite sally davis paul dollman (appointed 6 december 2016) neil king (appointed 5 april 2017) peter adams (alternate) nathan luckey mike parton (chairman) christian seymour / deepu chintamaneni (alternate) liliana solomon (chief financial officer) damian walsh group website independent auditors pricewaterhousecoopers llp, savannah house, 3 ocean way, southampton, united kingdom company 1 directors peter adams mark braithwaite deepu chintamaneni sally davis paul dollman (appointed 6 december 2016) neil king (appointed 5 april 2017) nathan luckey paul mullins (resigned 31 august 2017) mike parton christian seymour damian walsh company secretary michael giles registered ofiice crawley court winchester hampshire company registration number 05254001 annual report for the year ended 30 june 2017 1 in respect of arqiva group limited, the ultimate parent company of the group arqiva group limited cautionary statement this annual report contains various forwardlooking statements regarding events and trends that are subject to risks and uncertainties that could cause the actual results and financial position of the group to differ materially from the information presented herein. A reconciliation of the reported ebitda to the financial statements is provided below year ended 30 june 2017 £m year ended 30 june 2016 £m operating profit 284.5 271.1 depreciation 16 141.6 129.4 amortisation 15 12.6 10.4 exceptional items charged to operating profit 7 29.5 13.6 other income (1.1) (0.2) share of results of joint ventures and associates 17 (0.3) (0.1) other 2 0.2 0.2 1 ebitda is a non-gaap measure and refers to earnings before interest, tax, depreciation and amortisation' and includes add-backs for certain items charged to operating profit that do not reflect the underlying business performance. 29 arqiva group limited c.1150 tv transmission sites c.800 1 radio transmission sites 4 dtt multiplex licenses our customers include... business unit snapshot revenue £m £m headcount (ftes) 2015 2016 2014 393.6 2015 2016 2014 275.7 2015 2016 2014 592 2017 449.0 2017 329.4 2017 674 there was growth in t errestrial broadcast as a result of new channel launches and associated increases; utilising 31 videostreams on its main multiplexes and increasing channel sales on its dvb-t2 (hd enabled) multiplexes; dab roll-out and increased transmission activity thereon; increased activity in relation to the 700mhz clearance programme; and rpi-linked increases on broadcast service contracts. 4 limited united kingdom dormant company 30-jun 100% arqiva pension trust limited united kingdom dormant company 30-jun 100% arqiva pp financing plc united kingdom financing vehicle 30-jun 100% arqiva pte limited singapore satellite transmission 30-jun 100% arqiva public safety limited united kingdom dormant company 30-jun 100% arqiva sas france satellite transmission 30-jun 100% arqiva satellite limited united kingdom dormant company 30-jun 100% annual report and consolidated financial statements 2017 arqiva group limited (company reg 05254001) 133 company country of incorporation principal activities year end percentage of ordinary shares held arqiva senior finance limited united kingdom financing vehicle 30-jun 100% arqiva services limited united kingdom transmission services 30-jun 100% arqiva smart financing limited united kingdom financing vehicle 30-jun 100% arqiva smart holdings limited united kingdom holding company 30-jun 100% arqiva smart metering limited united kingdom smart metering communications 30-jun 100% arqiva smart parent limited united kingdom holding company 30-jun 100% arqiva srl italy satellite transmission services 30-jun 100% arqiva swing limited united kingdom dormant company 30-jun 100% (held directly) arqiva telecommunications asset development company limited united kingdom dormant company 30-jun 100%

Figure 6.4: Example summary from the GRU-base + attention model.

6.3 Future Work Mockflow

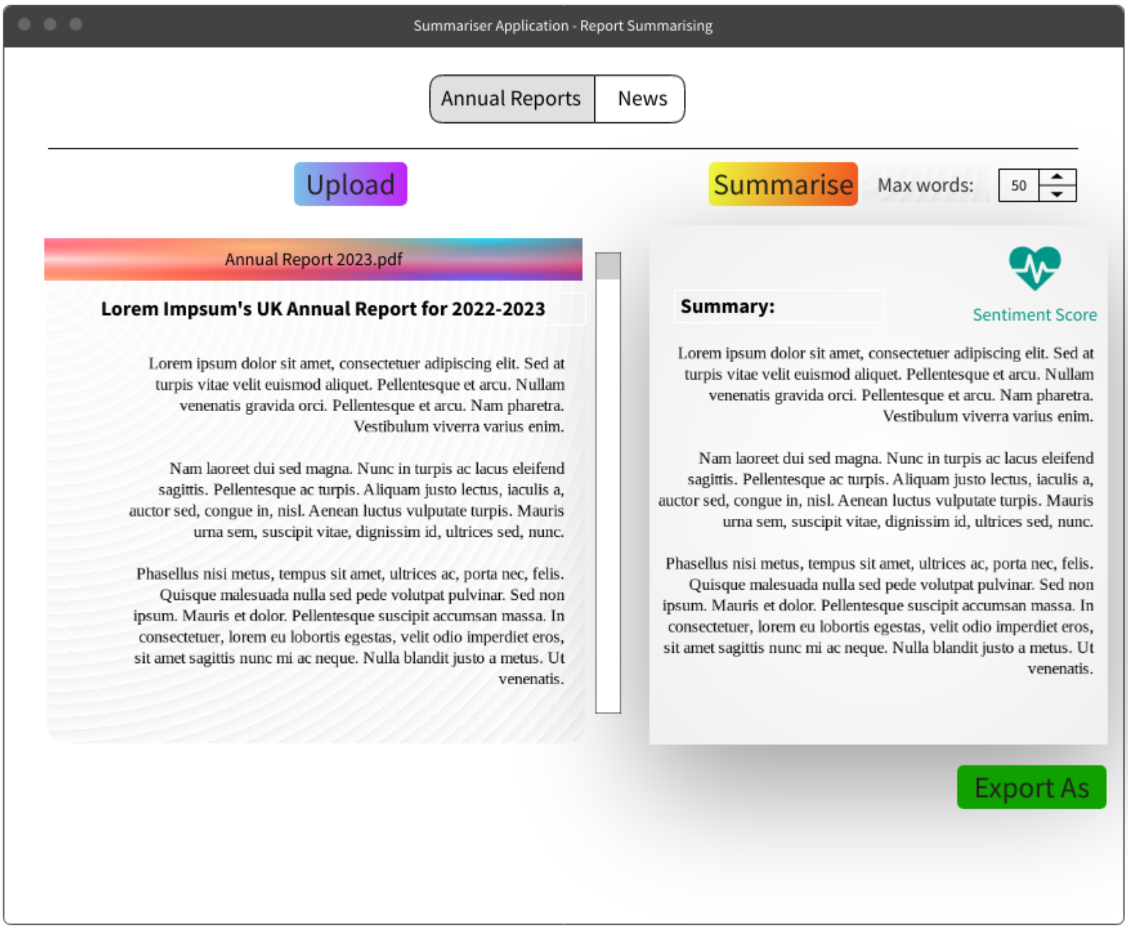


Figure 6.5: Report summarisation mockflow.

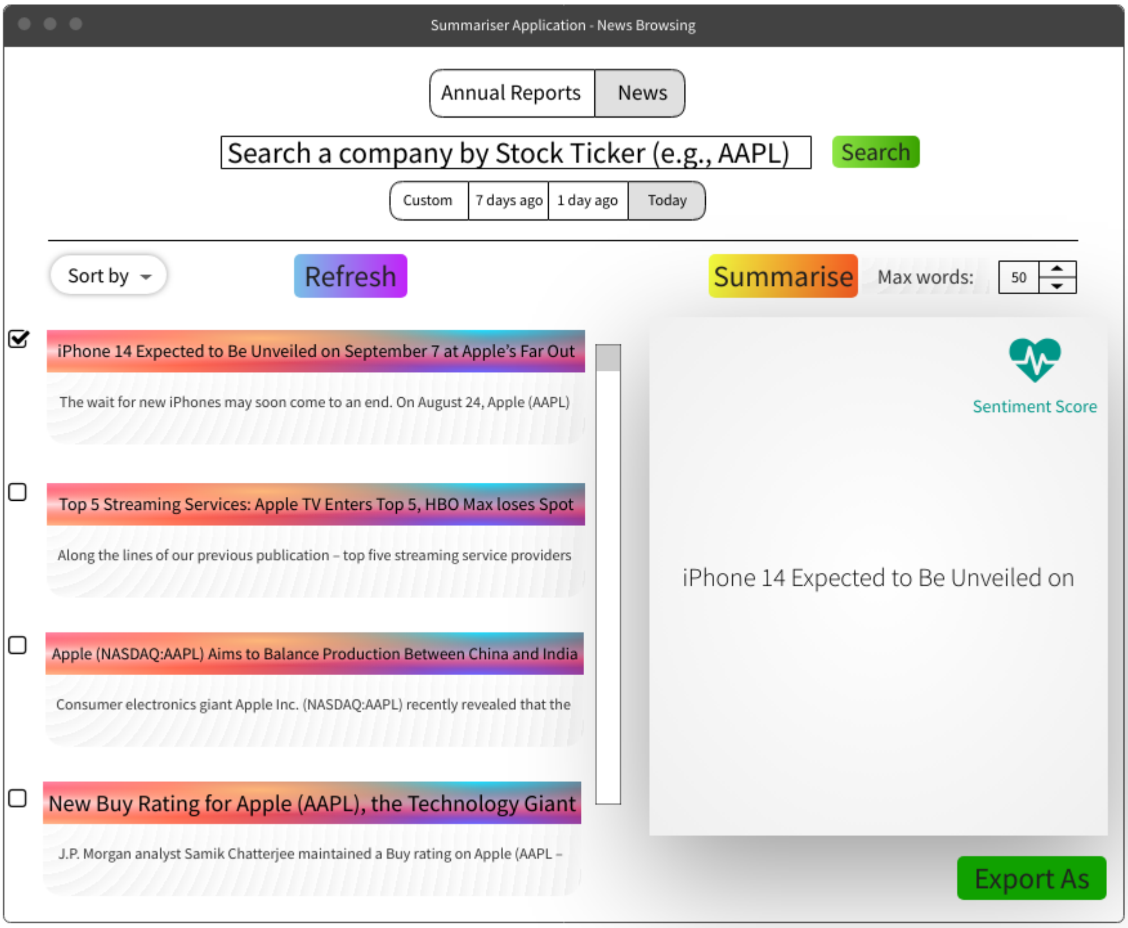


Figure 6.6: News browsing mockflow.