TEXT SUMMARIZATION & INFORMATION RETRIEVAL IN INSURANCE DOMAIN USING NLP

SHRADDHA MALADKAR

Final Thesis Report

MAY 2023

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor, Dr. Padmakar Pandey for his invaluable guidance and support throughout my research journey.

I am also grateful for all the online sessions, which provided me with valuable insights and enhanced my knowledge.

Additionally, I want to acknowledge the unwavering support of my family and friends, whose love and encouragement have been my driving force. Their contributions have been instrumental in the completion of my thesis.

ABSTRACT

Businesses are the backbone of an economy and rising competition is a fundamental driver of productivity and output growth. Competition can be considered healthy for business as it keeps the business on its toes and makes it imperative for it to innovate and improve. This creates a need for benchmarking the competition as benchmarking is a method of understanding company's performance against others and ensure that they can stay competitive in the market. There are different ways through which benchmarking can be performed like keeping an eye on competitor's social media, press releases, market trends and extract key information from all the data that is gathered by various sources. Traditionally, this is done via manual intervention that is reading and going through this huge amount of data, retrieving relevant information which is time consuming and prone to human error affecting the productivity of the business. Now these tedious tasks can be easily accomplished using NLP (Natural Language Processing) with techniques like Information Retrieval (IR) and Text Summarizer. Training models using these techniques makes productive summary generation probable and retrieve key information. There are two methods which are used commonly to generate summaries where extractive method scans the article to discover the related sentences while isolating only that piece of information from the article whereas abstractive technique understands the text and then generates the summary. Abstractive process is considered to be a bit more complex, hence transformer-based pretrained models are employed to compare the content & creating the summary. The study uses the data obtained by web scraping news articles and public disclosures of the competitors of Insurance domain to assess and compare the outcomes found using the machine learning models. The crucial information found out by the model can be then used to get insights into competitor's business and help the company determine the scope of improvement and devise strategies that work best for the business.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PR	Public Relations
HTML	HyperText Markup Language
XML	Extensible Markup Language
API	Application Programming Interface
SERP	Search Engine Results Pages
NLP	Natural Language Processing
IR	Information Retrieval
SDTS	Single Document Text Summarization
MDTS	Multi-Document Text Summarization
TF-ID	Term Frequency Inverse Document Frequency
HDFC	Housing Development Finance Corporation Limited
ICICI	Industrial Credit and Investment Corporation of India
SBI	State Bank of India
LIC	Life Insurance Corporation
GRU	Gated Recurrent Unit
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-term Memory network
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pretrained Transformer
T5	Text-To-Text Transfer Transformer
RoBERTa	Robustly Optimized BERT approach
ELMO	Embeddings from Language Models

BART	Bidirectional and Auto-Regressive Transformers
VQA	Visual Question Answering
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
NSP	Next Sentence Prediction
BLEU	Bilingual Evaluation Understudy
METEOR	Metric for Evaluation of Translation with Explicit Ordering
CIDEr	Consensus-based Image Description Evaluation
CSV	Comma-Separated Values
NLTK	Natural Language Toolkit
VoNB	Value of New Business

CHAPTER 1

INTRODUCTION

1.1 Background

As businesses experience growth and expansion, the market becomes increasingly competitive. In a rising business environment, it is essential for companies to keep pace with the market and effectively compete to ensure success. It is vital for businesses to recognize the importance of keeping up with market trends and effectively competing in order to thrive. In order to gain an edge over competitors and potentially even take over the market, it is crucial to closely monitor and track the competition. It is very essential to have a deep understanding of the competitive landscape. Competitive intelligence plays a pivotal role in gaining this understanding.

Before the internet was discovered, competitive intelligence was majorly conducted through industrial espionage. While the latter still occurs, there are much simpler and more ethical ways to gain insights on the competitors' business. By being attentive to the data competitors and their customers leave on the internet, the company will be able to make well-informed strategic decisions.

Competitive intelligence also known as market intelligence is the action of gathering and analysing information about products, market, customers and competitors. Through this method, one can expect competitor's likely next steps, spot possible threats & opportunities, and gain competitive benefit. By collecting this valued information through competition benchmarking makes it easier for the company to make an effective and competent competitive strategy (Kuhanec T., 2022).

Benchmarking is a way of figuring out company's accomplishments against others and more importantly guarantees that they can stay ahead in the market. It offers an insight into what a competitor is achieving, what works, what doesn't work, and something that companies might use for their own progress, whether it's achieved by adopting effective practices or dodging ineffective ones.

The reason why competitive benchmarking is so significant is because it helps businesses understand their position within the competitive landscape, recognize opportunities, identify

their market share along with many challenges in the industry. And to do this, a brand needs to understand how they measure up against the market and collect some data on other brand's strategies and with no helpful way to contextualize the data, the company is stuck inspecting data in vain. This results in lagging of company's social strategy thus proving ineffective.

Traditionally, competitive benchmarking is very long and laborious process which is particularly time consuming for teams to implement and frequently found with human errors. One needs some data-driven insights or else it's very tough to quantify what accomplishment looks like compared to other companies in the market or to get a clear picture on top-notch trends. And if real-time data is missing, one can neither form effective strategies nor can they advise customer care services, social media, PR teams and marketing teams (Sprinklr Team, 2021). Now, this competitor's data can be extracted from various sources like news article, social media posts, press releases, official product launches and public disclosures using techniques like web scraping.

Web scraping is method that can automatically extract huge amount of data from any web pages. Majority of the extracted data is unorganised and in HTML format which can be transformed into structured data, stored into a database and can be made useful in many applications. There are numerous ways to do web scraping such as using software that are readily available online, particular API's or even generating a code for web scraping using any language like Python, Java etc. Some websites including Twitter, Facebook, Google etc. exposed their API which permits any user to gain access of the data in organized format. But there are some sites that blocks user's attempts to gain access to huge data available in organized format or they may lack technologically advancement. Such situations arise a need to scrape data from any websites.

Once the data is gathered which is usually huge in size needs to be processed and then analyse to draw insights and use it to the company's benefit. This is where NLP techniques like Information Retrieval (IR) and Text Summarization comes into picture.

Information retrieval is the process of retrieving the most accurate information from text which is usually based on a specific query provided by the user using context-based indexing or metadata.

1.2 Text Summarization and Types

NLP text summarization is the method of breaking down long texts into consumable paragraphs or sentences. This method separates vital information while conserving the meaning of the text. This decreases the time required for grasping prolonged pieces such as articles without the loss of any crucial information.

Types of Summarization techniques are based on input type, output type and purpose of summarization.

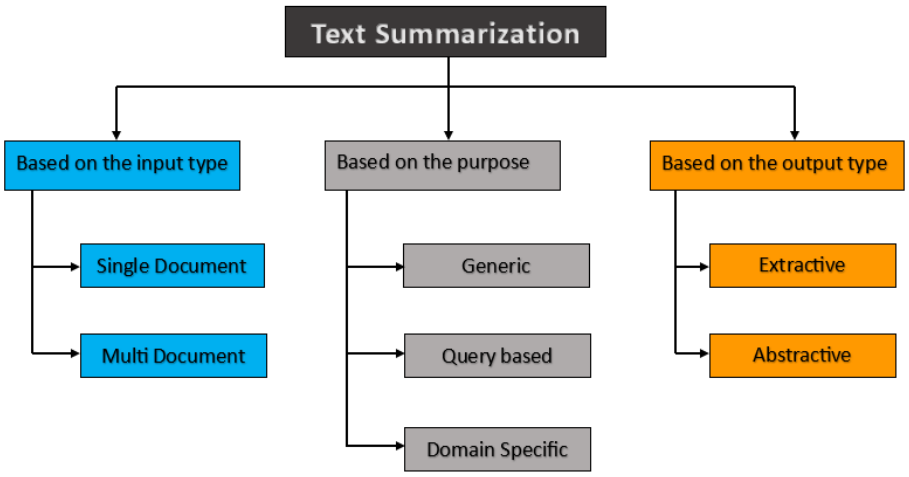


Figure 1.2. Types of Text Summarization

1.2.1 Summarization based on input type

1.2.1.1 Single Document Text Summarization (SDTS)

Input length is short in SDTS type of Summarization as input contains only single document. This was widely used in the early days of Text Summarization.

1.2.1.2 Multi-Document Text Summarization (MDTS)

Multi-document text summarization is a method employed when the input length on a specific topic is extensive, necessitating the use of multiple documents as input for the summarization process. Unlike single document text summarization, this technique presents challenges due to the need to consolidate summaries from various documents into a cohesive whole. The difficulty lies in reconciling the diverse themes found across different documents. An optimal

summarization technique must effectively condense the primary themes while ensuring readability, comprehensiveness, and the inclusion of crucial sentences.

1.2.2 Summarization based on output type

1.2.2.1 Extractive Technique

In extractive summarization, from the text fed as the input, a summary is produced by choosing a subsection of the entire sentence. After this, the most significant phrases or sentences are recognized and chosen on basis of a score that is calculated based on the words present in that sentence.

1.2.2.2 Abstractive Technique

In this approach to summarization, the focus is on providing an abstract summary of an article by extracting the most crucial facts from the text. The abstractive summarizer begins by comprehending the primary concepts within the document and then articulates those concepts using fewer words. Instead of relying solely on copying and rephrasing sentences from the source text, this method aims to capture the essence of the content and present it in a more concise and condensed manner. By understanding the core ideas, the abstractive summarizer effectively communicates the key information while utilizing a reduced word count. It generates one-of-a-kind summary.

1.2.3 Summarization based on purpose

1.2.3.1 Generic Text Summarization

This refers to a method where the model does not make any inferences about the meaning of the text or possess domain-specific knowledge. It generates a generic summary that encompasses the entire text, including documents, photos, or video clips. By disregarding contextual or domain-specific information, this technique aims to provide a broad and generalized summary that captures the overall content without delving into specific details or nuances. This allows for a more universal summarization process that can be applied to various types of text-based content, regardless of the subject matter or domain.

1.2.3.2 Query-based Text Summarization

In this method, the summarization process revolves around utilizing a query as input, which guides the model in selecting relevant sentences and phrases to create a summary. By considering the provided query, the model identifies and extracts the portions of the text that are closely tied to the query's content. This targeted approach allows the model to generate a summary specifically tailored to the query, ensuring that the resulting summary is focused on the aspects directly associated with the query's subject matter. By leveraging the query's context, the summarization model can effectively capture and highlight the most pertinent information in the summary.

1.2.3.3 Domain-Specific Text Summarization

This approach to text summarization involves leveraging domain-specific knowledge, such as scientific or medical expertise. By incorporating this specialized knowledge, the summarization process attains heightened accuracy, resulting in more meaningful, concise, and easily comprehensible summaries of the entire text. The utilization of domain-specific knowledge empowers the model to capture the intricacies and nuances unique to the chosen domain. Consequently, this approach enhances the quality and relevance of the summaries, making them highly informative and accessible to readers.

1.3 Problem Statement

The amount and difficulty of unstructured data is increasing exponentially and this brings forward some new challenges. More the time spent on manual data extraction means less time spent on analysis which creates delays in drawing insights and informing those insights to stakeholders. Also, there is always a possibility of human error or poor data quality with doing things manually.

Currently gaining information on competitors is done via visiting their official websites, finding documents, press releases, news articles and social media presence featuring the competitors. One needs to pro-actively keep an eye on competitor's performances, product launches, news updates as well as keep track of market trends, manually collect data from multiple sources and for all the competitors that are present in the market. Then information is extracted from such documents/news articles/social media posts by reading each and every line and then it is summarized by an individual.

This manual intervention makes this process vulnerable to human error affecting the accuracy of the results. There are chances where some of the information might be overlooked which would have been important for business.

To overcome above mentioned challenges, I am proposing a system with -

- **Web Scrapper tool** which will be used to gather the competitor's data and market trends using Web Scraping which will help us gain real time data.
- **Text Summarization & Information Retrieval tool** which will extract relevant and important information and also condense the news articles/ social media posts into a shorter version and preserve important contents

1.4 Aim & Objectives

The aim of this research is to build an information retrieval tool that extracts key information form the documents and a text summarization tool which focuses on the issue of identifying the most significant parts of the text and generating coherent summaries from the news articles.

The goal of this study is to create a social listening tool that monitors competitor's actions and provides only vital information which can in turn help devise effective strategies and achieve optimal growth of the business.

The research objectives are formulated based on the aim of this study, which are as follows:

1. To collect data using Web Scraping, combine the data extracted from multiple sources and perform data cleaning operations

2. To implement Information Extraction technique to extract information from the documents

3. To create an optimized and efficient algorithm in order to produce a text summary of the news articles

4. To evaluate the performance of the techniques/algorithms to achieve optimal result

1.5 Significance of the Study

There is an exponential growth in the daily generated data as the cost of storing the data is reducing and computational power of the systems is increasing. But any means to extract the information and be able to query it is lacking then the information collected is useless. Information is knowledge only if we are able to extract relevant information. It takes time as well as efforts to read an entire article, analyse it, and bifurcate the important concepts from the raw text. At least 15 minutes are required to read a 500-word article while automatic summary tool summarises texts of 500-5000 words in mere seconds. This allows the user to read less data and still be receiving the key information and making sound judgments. Instead of reading entire news stories that most of the time contains unrelated information, summaries of such websites can be accurate with only about 30% of the original article's size. These tools in turn can help companies in numerous ways, elimination of manual intervention to extract data/information will provide more accurate data with reduced amount of time. This reliable data will help in making strategies and taking business decisions which can lead to greater

profits. Keeping track of the competitor's action will benefit in finding drawbacks in their own products/services and improve the quality of the same.

1.6 Scope of the Study

The scope determines the level up to which the research can be discovered and how the methods/techniques can be bound to specific constraints. The study will utilize web scraping techniques to collect news articles from the competitors. By leveraging NLP techniques, various methods and algorithms will be explored to identify the most effective models. To accurately assess the model's performance, the study will employ the ROUGE metric, ensuring a reliable measure of its accuracy.

1.7 Structure of the Study

Chapter 2 provides an in-depth exploration of the extensive literature on Text Summarization, tracing its evolution from early approaches such as TF-ID and lexical chains to the adoption of more advanced techniques like RNN, LSTM. It further highlights the revolutionary impact of transformers and pre-trained models on the field. In Chapter 3, the selection and preparation of the dataset for the Study are elaborated, along with the rationale behind choosing a specific dataset. Furthermore, the section encompasses an exploration of diverse methodologies and model architectures that can be used for the Text Summarization and Information Retrieval system task. Chapter 4 presents the practical implementation of the selected and refined methodology for the tasks at hand, accompanied by a detailed description of the data pre-processing steps. Chapter 5 of the study focuses on the evaluation of the model's performance and results. Various techniques, including ROGUE, are employed to assess and analyse the effectiveness of the model in achieving the desired objectives. Chapter 6 concludes the Study and provides future scope and recommendations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In Natural Language Processing, text summarizer is a vast field of research and many previous research works have contributed towards automatic text summarization.

2.2 Origin of Text Summarization

Introduction to Natural Language Processing and Text Summarization was first done in 1958 by (H. P. Luhn, 1958). His approach involved analysing the structural and semantic features of scientific articles to extract key sentences that encapsulated the main ideas. He developed a set of heuristics and algorithms to identify important sentences based on factors such as word frequency, position, and grammatical structure.

The method began by selecting sentences that contain frequently occurring words or phrases, considering them as potential candidates for summarization. These sentences were then ranked based on their position within the document, with sentences closer to the beginning or end given higher priority. The researcher further introduced the concept of significant words, which are terms that occur frequently within the article but had limited occurrence in the overall literature. Sentences containing these significant words were deemed more informative and were more likely to be included in the generated abstract.

Initially, statistical approaches were explored where a score was computed for every sentence and then the sentences were selected with the highest scores disregarding the very high frequency common words. Various techniques were used to calculate this score, such as TF-IDF, Bayesian models, etc. (H. P. Edmundson, 1969) introduced a novel method called "TF*IDF" (Term Frequency * Inverse Document Frequency) for ranking the importance of sentences within a document. This method assigned higher weight to sentences which contained rare terms that were specific to the document, thus identifying sentences with unique and significant information. To further enhance the extraction process, the author proposed the use of additional criteria to filter sentences. These criteria included sentence position, cue words, and thematic words. Sentences that appeared at the beginning or end of a document, contained

specific cue words, or were related to the document's main theme were given higher relevance scores.

2.3 Lexical chains Evolution

The concept of lexical chains was first introduced by (Morris and Hirst.,1991) where he argued that traditional methods, such as sentence-level analysis, failed to capture the subtle connections between words and phrases that contributed to the overall coherence of a text. To address this issue, the authors introduced the concept of lexical chains as a means of representing the semantic relationships between words throughout a document. A lexical chain is a sequence of related words that share a common semantic thread. These chains served as indicators of the underlying coherence and thematic structure of the text. Morris, Jane, and Hirst proposed a systematic approach to identify and construct lexical chains. The process involved examining the relationships between adjacent words in a text and determining if they form a cohesive chain based on specific criteria. These criteria included similarity of meaning, repetition, and word association.

(Barzilay and Elhadad, 1999) explored the application of lexical chains, using same concept introduced by Morris, Jane, and Hirst, to the task of text summarization. The researchers presented a detailed methodology for constructing lexical chains and incorporating them into the summarization process. Their approach involved identifying content words, building chains based on word associations, and ranking sentences based on the presence and quality of the chains they contained.

(Silber and McCoy, 2002) argued that traditional methods, such as lexical cohesion and semantic networks, often suffered from high computational costs or failed to capture the underlying semantic structure effectively.

To overcome these challenges, the authors introduced an algorithm that efficiently computed lexical chains based on the concept of lexical cohesion. Their method involved scanning the text and identifying chains of related words that shared a common topic or theme. The algorithm utilized a table-driven technique, making it computationally efficient even for large texts.

Lexical chains rely heavily on identifying and connecting words based on their lexical relationships. However, this approach can be limited in capturing the full semantic meaning of the text. It may not consider the context or deeper understanding of the content, leading to

potential inaccuracies in the summary. Lexical chains primarily focus on identifying related words within the text but may not adequately consider the broader context or external knowledge. This limitation can result in a summary that lacks essential background information or fails to capture the nuances and subtleties present in the original text. Text summarization based on lexical chains can be sensitive to noise or errors in the input text. To overcome these limitations, researchers have explored alternative approaches such as extractive summarization based on machine learning and neural networks, which aim to capture a deeper understanding of the text and produce more coherent and contextually accurate summaries.

2.4 Exploring Extractive Summarization

(Chatterjee et al., 2012) proposed a Text Summarizer based on Genetic Algorithms, utilizing a Directed-Acyclic-Graph representation of a document. The researchers introduced a schema where each edge of the graph was assigned a weight, aiming to maximize an Objective function. This function considered factors like readability, sentence cohesion, and topic relevance. The Genetic Algorithm was employed to select key sentences from the text by evaluating the Cohesion Factor and assigning a higher preference to sentences related to the input query (Topic Relation Factor). The Objective function's maximization through the Genetic Algorithm ultimately produced an optimized summary.

A multi-document summarization technique was proposed by (Tandel et al., 2016) which allowed the used to shorten relevant data from multiple documents provided as a single input. A lot of time was saved with increased efficiency using this method. They were inspired from the then existing methods like Cluster-based, Lexical Chain based and Topic-based.

The paper titled (Patel et al., 2017) presented an overview of various techniques and algorithms used in extractive summarization and highlighted their advantages and limitations. The authors presented a survey of extractive-based techniques, categorizing them into three main groups: statistical methods, graph-based methods, and machine learning approaches.

Statistical methods rely on statistical measures such as term frequency, sentence position, and sentence length to determine the importance of sentences. Algorithms like TF-IDF, frequency-based ranking, and sentence position-based ranking fall under this category.

Graph-based methods represent the document as a graph, with sentences as nodes and edges representing relationships between them. TextRank and LexRank are algorithms which use graph-based structures to rank sentences based on their importance and coherence.

Machine learning approaches utilize supervised or unsupervised learning algorithms to train models on a corpus of documents and their corresponding summaries. Algorithms such as support vector machines, k-means clustering, and neural networks can be employed in extractive summarization.

Drawbacks of Extractive Summarization-

Contextual Understanding: They fail to capture the underlying meaning and generate summaries that go beyond the exact wording of the original sentences as they rely solely on selecting and rearranging existing sentences, which may limit its ability to capture the complete essence of the text.

Information Compression: Extractive Summarization is limited to using sentences from the source text, which may result in longer summaries and the inclusion of less significant information also limiting the flexibility and creativity of the generated summaries.

Handling Complex Texts: Abstractive summarization is better suited for complex texts that contain specialized vocabulary, jargon, or intricate structures. It can understand and simplify complex concepts, making the summary more accessible to a wider audience. Extractive summarization may struggle to handle such complexities, potentially leading to summaries that are less comprehensive or coherent.

Adaptability to New Information: Abstractive summarization models have the potential to generate summaries for content that doesn't have prior summaries available. They can adapt to different domains and handle new or unseen information effectively. Extractive summarization relies on the availability of pre-existing summaries or requires updating the summary with new information manually.

All of these techniques generated a good summary using key phrase extraction but these approaches were extractive and as a result were chopping off the original text which led to some information loss. These limitations can be overcome by using Abstractive Text Summarization techniques as goes beyond the selection and rearrangement of sentences from the original text and aims to generate a concise summary that captures the key information in a more human-like manner.

2.5 Power of Sequential Models: Exploring RNN, LSTM, and Beyond

When we read or write, the order of words in a text or sentence matters a lot for understanding its meaning. This made text and sentences similar to a sequence of information. For many problems in natural language processing (NLP), there was a need to handle this sequential data. To do that effectively, the architecture or structure of NLP models needed to be able to remember important details using something like a memory. This helped the model process and interpret text in a way that considered the order of words, allowing it to understand and generate more accurate and meaningful language.

(Hochreiter et al., 1997), published a ground breaking recurrent neural network (RNN) architecture known as Long Short-Term Memory (LSTM). The authors addressed the vanishing and exploding gradient problems that hinder the training of traditional RNNs, making them unsuitable for capturing long-term dependencies. They discussed the limitations of standard RNNs and the challenges they face in preserving and propagating relevant information over long sequences. The authors proposed the LSTM architecture as a solution, which incorporates memory cells and gating mechanisms to selectively retain or forget information at different time steps.

The core components of the LSTM are the memory cell and three types of gated units: input gate, forget gate, and output gate. These three gates handle the flow of data into and out of the memory cell, allowing the LSTM to capture and preserve long-term dependencies while avoiding the issues of vanishing and exploding gradients.

The paper's contributions had a profound impact on the field of deep learning and paved the way for advancements in various domains involving sequential data processing.

(Nallapati R et al., 2016) published a paper introducing a novel approach to text summarization using sequence-to-sequence recurrent neural networks (RNNs). The authors aimed to tackle the limitations of traditional extractive summarization methods by employing a more flexible and expressive abstractive approach. The proposed model was built upon the sequence-to-sequence architecture using an encoder-decoder framework. To improve the performance, hierarchical attention mechanism was utilized that enabled the model to attend to relevant parts of the source text at different levels of granularity. This attention mechanism helped in order to capture important information and improve the overall quality of the generated summaries. Seq2Seq models relied on a predefined vocabulary, and any out-of-vocabulary (OOV) words in the input or during decoding posed a challenge. Also, Seq2seq models employed using encoder-decoder

framework in order to solve NLP tasks provided with wonderful results, but there was still the problem of parallelization. Even if the sequential information is retained in the case of the encoder-decoder model, the processing was executed by taking one input at a time as LSTM allows only a single input at a given time. Hence, the sequential nature of RNNs limits parallelization during training and inference, making them slower compared to other architectures such as transformers.

The paper by (Sneha Choudhary et al., 2020) proposed a document retrieval system using deep learning techniques. They combined the traditional TF-IDF approach with Bidirectional Encoder Representations from Transformers to create semantically rich document embeddings. By incorporating contextual embeddings, BERT addressed the limitations of TF-IDF in capturing semantic context. The authors developed an ensemble model that ranked documents based on the weighted sum of TF-IDF and BERT scores. The model was evaluated on MS MARCO data and showed significant performance improvements compared to the TF-IDF method.

2.6 The Rise of Transformers: Revolutionizing Natural Language Processing

The paper titled "Attention is All You Need" by (Vaswani et al., 2017), presented a breakthrough model called the Transformer. The Transformer architecture based solely on self-attention mechanisms replaced traditional recurrent and convolutional neural networks in many natural language processing tasks.

The author presented the limitations of previous sequence transduction models that heavily relied on recurrent or convolutional layers. He also claimed that these models suffer from slow training, difficulties in parallelization, and an inability to capture long-range dependencies effectively.

(Vaswani et al., 2017) proposed the Transformer as a solution, which was built on the concept of self-attention mechanisms. The model employed attention mechanisms to directly relate different positions within a sequence to compute representations of each word in the sequence. This attention-based approach allowed for parallel computation and captures global dependencies more efficiently.

To address the limitations of Seq2Seq transformers, researchers explored not only alternative architectures, such as transformers but also their variants (e.g., BERT, T5), which showed

significant improvements in text summarization tasks by capturing long-range dependencies, incorporating self-attention mechanisms, and providing better context understanding.

(Devlin J et al., 2018) introduced a ground breaking language representation model called BERT (Bidirectional Encoder Representations from Transformers). This transformer revolutionized the field of natural language processing by significantly improving language understanding tasks. The researchers proposed BERT as a solution by leveraging the power of transformers, and introduced the concept of pre-training and fine-tuning. They introduced a next sentence prediction task to train the model to understand the relationship between two sentences.

BERT uses a bidirectional approach, allowing the model to capture contextual information from both left and right context, which leads to a better understanding of words and their surrounding context.

Huggingface developed a Transformers library which is built on PyTorch and TensorFlow frameworks introduced by (Wolf et al., 2019), provided a comprehensive and efficient implementation of state-of-the-art models for NLP tasks, including both pre-trained models and tools for fine-tuning. The library provided a unified and user-friendly interface for various transformer-based models, such as BERT, GPT, and RoBERTa. The library has support for various model architectures, pre-trained weights, tokenizers, and fine-tuning tools. The library offers a wide range of transformer-based models, pre-trained weights, and tools for fine-tuning, providing researchers and practitioners with a powerful and accessible platform for NLP tasks. The benchmark results and community contributions validate the effectiveness and impact of the Transformers library in advancing the field of NLP.

2.7 Summary

To summarize the chapter, the field of text summarization has evolved significantly over the years. It began with H. P. Luhn's approach of extracting key sentences from scientific articles. Overtime techniques such as statistical methods, lexical chains, and extractive summarization were explored. Sequential models like LSTM and transformative architectures like the Transformer revolutionized the processing of sequential data and language understanding. The introduction of BERT and the Transformers library provided powerful tools for NLP tasks which will be explored in chapters ahead.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The research methodology focuses on exploring different approaches for text summarization and delving into the underlying architecture of each approach. It aims to identify the data requirements based on the problem statement and select appropriate data sources. Furthermore, the methodology aims to devise methods to extract the necessary data for the study.

3.2 Data

To effectively gather competitive intelligence on competitors, it is crucial to obtain reliable, relevant, genuine and precise data. By extracting actionable information from this data and deriving valuable insights, businesses can drive growth and stay ahead in the market. For this study, news articles and public disclosures of companies like HDFC Life, ICICI Prudential, LIC, SBI Life, Max Life etc have been chosen as the primary data sources.

3.2.1 Data selection

News articles are often a rich and valuable source of information for gaining competitive intelligence. Following are some reasons why news articles were chosen as a data source:

1. Timeliness: News articles provide real-time and up-to-date information about competitors, their activities, product launches, market trends, and industry developments. This timeliness allows businesses to stay informed about the latest competitive landscape.

2. Broad Coverage: News articles cover a wide range of topics, including industry news, market analysis, financial reports, mergers and acquisitions, strategic partnerships, and more. This breadth of coverage enables businesses to gather comprehensive insights about their competitors' strategies, initiatives, and performance.

3. Publicly Available: News articles are generally accessible to the public, making them a convenient and readily available source of information. This accessibility enables businesses of

all sizes to access and leverage news articles for competitive intelligence without significant barriers.

4. Unbiased Information: While it's important to critically evaluate the credibility of news sources, reputable news articles often strive to provide objective and unbiased information. This can help businesses obtain reliable and neutral insights about their competitors, minimizing the risk of biased or skewed perspectives.

By leveraging news articles as a data source for competitive intelligence, businesses can stay informed, make informed decisions, and adapt their strategies in response to market dynamics and competitor actions.

Public disclosures refer to the information that insurance companies are legally required to make available to the public and regulatory authorities. Public disclosures provide detailed and reliable information about competitors' financial performance, including their revenue, profit margins, loss ratios, and investment portfolios. Analysing this data allows businesses to assess the financial health and stability of their competitors, gain insights into their profitability, and identify potential areas of competitive advantage. By leveraging this data, businesses can gain insights into competitors' financial performance, market strategies, and risk profiles. These insights enable businesses to make informed decisions, refine their own strategies, and identify opportunities for growth and competitive advantage. Public disclosures include-

1. Financial Statements: This includes details of the company's financial performance, including balance sheets, income statements, and cash flow statements.

2. Annual Reports: These reports provide an overview of the company's operations, financial highlights, corporate governance practices, and strategic initiatives.

3. Regulatory Compliance: Information related to compliance with regulatory requirements imposed by insurance regulators or other relevant authorities.

4. Risk Management: Details about the company's risk management framework, including risk assessment methodologies and strategies to mitigate risks.

5. Investments: Information on the company's investment portfolio, including asset allocation, investment strategies, and performance.

6. Policyholder Protection: Disclosure regarding measures taken by the company to protect policyholders' interests and ensure fair treatment.

7. Solvency and Capital Adequacy: Data related to the company's solvency position and capital adequacy, ensuring its ability to meet policyholder obligations.

8. Business Operations: Details about the company's business model, distribution channels, product offerings, and market segments.

9. Key Performance Indicators: Metrics and ratios that provide insights into the company's performance and financial health.

Thus, news articles and public disclosure data have been leveraged in this study to conduct competition benchmarking among insurance companies.

3.2.2 Data Collection

Since a readily available dataset containing news articles in the insurance domain is not accessible, manual data collection becomes necessary to create a dataset for the study.

Web scraping presents a viable solution for extracting news articles related to the insurance domain by utilizing competitor names as keywords to search for relevant articles on Google. Google offers an API that enables users to retrieve a large amount of data by making API calls through a script written in Python or another programming language. This approach allows for efficient and automated extraction of data from Google search results.

In the script, web scraping libraries can be utilized to write the code and automate the data extraction process. Libraries such as BeautifulSoup and Scrapy in Python provide powerful tools for parsing HTML or XML content and extracting the desired information from web pages. By integrating these libraries, user can navigate through the search results page, locate the relevant news articles, and extract the necessary data such as headlines, summaries, publication dates, and article texts. The web scraping libraries provide functions and methods to access specific HTML elements, filter and extract data based on defined patterns, and handle pagination if required. The code can be customized to use competitor names as search keywords and iterate through multiple search result pages to gather a comprehensive dataset of insurance-related news articles.

Public disclosure data files can be downloaded from official websites of insurance companies as this data is made readily available to public in order to maintain transparency.

3.3 Text Summarization Strategies

Text summarization involves processing a collection of sentences, which can be referred to as sequence data due to their ordered nature. In this type of data, the order of observations is significant, as earlier sentences provide information that is relevant to later ones, and vice versa. To effectively process and understand this sequential information, specialized models like Seq2Seq and Transformers have been developed and implemented. These models are specifically designed to handle the complexities and dependencies present in sequential data for tasks such as text summarization,

The decision to use sequence models in this study was driven by their capability to effectively capture the intricacies of sequential information, enabling accurate and concise text summarization.

3.3.1 Understanding Sequence Model (Seq2Seq)

A Seq2Seq model is an encoder-decoder based model implemented to handle such sequence data that takes a sequence of items such as words and outputs another sequence of items.

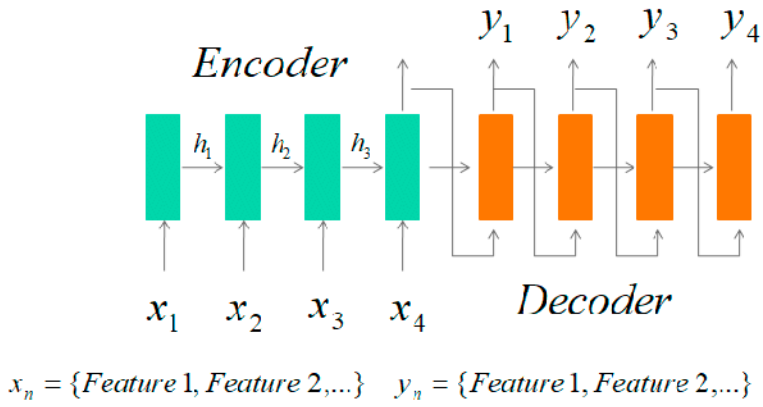


Figure 3.3. Seq2Seq Model Architecture

The encoder takes an input sequence, such as a sentence in the source language, and processes it into a fixed-dimensional vector representation, called the context vector or the encoder hidden state. The encoder can be a recurrent neural network (RNN), such as a Long Short-Term Memory (LSTM) or a Gated Recurrent Unit (GRU), which processes the input sequence one element at a time, updating its hidden state along the way.

The decoder, which is another RNN, takes the context vector generated by the encoder and generates an output sequence, such as a translated sentence in the target language. In traditional Seq2Seq models, the decoder receives only the context vector at the beginning and uses it as the initial hidden state. However, this approach suffers from a limitation: it forces the decoder to compress all the relevant information of the input sequence into a fixed-length vector.

3.3.1.1 Attention Mechanism

The attention mechanism addresses this limitation by allowing the decoder to focus on different parts of the input sequence dynamically, or attend to different elements of the source sequence while generating each element of the target sequence. Instead of relying solely on the fixed-length context vector, the attention mechanism introduces attention weights that indicate the importance of different parts of the input sequence at each decoding step.

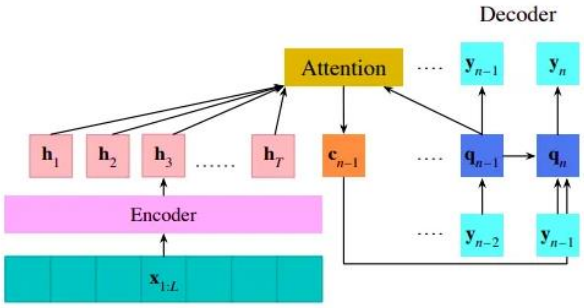


Figure 3.3.1. Attention Mechanism

1. During the encoding phase, the encoder processes the input sequence and produces a sequence of encoder hidden states.
2. At each decoding step, the decoder generates a decoder hidden state based on the previously generated elements of the target sequence.
3. The attention mechanism computes a set of attention weights by comparing the current decoder hidden state with the encoder hidden states. The attention weights represent the relevance or importance of each encoder hidden state for the current decoding step.
4. The attention weights are used to compute a weighted sum of the encoder hidden states, producing a context vector. The context vector is a weighted combination of the encoder hidden states, where the weights are determined by the attention weights.

5. The context vector is then combined with the decoder hidden state and fed into the decoder to generate the next element of the target sequence.

By incorporating the attention mechanism, the decoder has the ability to focus on different parts of the input sequence at different decoding steps, allowing the model to capture the relevant information more effectively. This results in improved performance where the input and output sequences can have different lengths and require more flexible alignment.

3.3.1.2 Advantages of Seq2Seq Models

1. Versatility: Seq2Seq models excel at a wide range of tasks, including machine translation, text summarization, image captioning, and more. They can handle variable-length input and output sequences, making them highly flexible.

2. Sequential Data Processing: Seq2Seq models are particularly effective when working with sequential data types like natural language, speech, and time series data.

3. Contextual Understanding: The encoder-decoder architecture of Seq2Seq models enables them to capture the context of the input sequence and utilize it for generating the output sequence.

4. Attention Mechanism: By incorporating attention mechanisms, Seq2Seq models can focus on relevant parts of the input sequence during the output generation process. This attention mechanism is especially beneficial for longer input sequences.

3.3.1.3 Disadvantages of Seq2Seq Models

1. Computational Intensity: Training Seq2Seq models demands substantial computational resources, making them computationally expensive. Optimization can also be challenging.

2. Limited Interpretability: Understanding the internal workings of Seq2Seq models can be difficult, hampering interpretability and making it challenging to explain model decisions.

3. Overfitting Risk: Seq2Seq models are prone to overfitting if not properly regularized. This can lead to subpar performance when working with new, unseen data.

4. Handling Rare Words: Seq2Seq models may struggle with rare words that were not present in the training data, potentially impacting their performance and output quality.

5. Handling Long Input Sequences: Seq2Seq models may encounter difficulties when processing very long input sequences, as the context vector might struggle to capture all the necessary information from the input sequence.

3.3.2 Overview of Transformer Architecture

Seq2seq models are a general framework for mapping an input sequence to an output sequence, typically consisting of an encoder and a decoder. Seq2seq models traditionally use recurrent neural networks (RNNs) as their building blocks, such as LSTM or GRU.

Transformers, on the other hand, are a specific type of seq2seq model architecture that rely on self-attention mechanisms. Unlike RNN-based models, transformers do not process the input sequence sequentially. Instead, they operate on the entire sequence in parallel, using self-attention to capture dependencies between all input positions. This allows transformers to capture long-range dependencies more effectively, making them well-suited for tasks involving long sequences, such as machine translation and text summarization. The Transformer model is a neural network architecture introduced in the paper "Attention Is All You Need" by Vaswani et al. It has revolutionized various natural language processing tasks by addressing the limitations of traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

The key components of transformers are multi-head self-attention layers and position-wise feed-forward networks. Self-attention enables the model to weigh the importance of different positions in the input sequence when generating each output position, while the feed-forward networks provide non-linear transformations to the representations.

As transformers have several advantages over traditional seq2seq models, this architecture was preferred for this study for text summarization. They can capture both local and global dependencies in the input sequence, handle long-range dependencies more effectively, and parallelize computations, making them more efficient for training on modern hardware.

3.3.2.1 Key Aspects of Transformers

Transformers have revolutionized Natural Language Processing (NLP) with their remarkable performance and capabilities. Here are some key aspects of Transformers in NLP:

1. Attention Mechanism: Transformers rely on self-attention mechanisms to capture dependencies between words in a sequence. This attention mechanism allows the model to focus on relevant parts of the input sequence, enabling better understanding and context modelling.

2. Parallelization: Unlike traditional recurrent neural networks (RNNs), Transformers can process inputs in parallel, making them highly efficient for training and inference. This parallelization significantly accelerates the processing time and allows for better scalability.

3. Positional Encoding: Transformers utilize positional encoding to incorporate positional information into the input embeddings. This encoding enables the model to capture the sequential order of words without the need for recurrent connections, which is a key advantage over RNNs.

4. Pretrained Language Models: Transformers have facilitated the development of powerful pretrained language models like BERT, GPT and T5. These models have achieved state-of-the-art results in various NLP tasks, including text classification, named entity recognition, sentiment analysis, and question answering.

5. Transfer Learning: Transformers enable transfer learning in NLP. Pretrained models can be fine-tuned on specific downstream tasks using a relatively small amount of task-specific labelled data. This transfer learning paradigm has proven effective in improving model performance, reducing training time, and addressing data scarcity issues.

6. Long-Term Dependencies: Transformers excel at capturing long-term dependencies in sequences. Unlike RNNs, which suffer from the vanishing or exploding gradient problem, Transformers can effectively model dependencies between distant words, making them more suitable for tasks that involve long-range dependencies.

7. Multimodal Applications: Transformers have been extended to handle multimodal data, such as combining visual and textual information in tasks like image captioning, visual question answering (VQA), and image generation. By leveraging self-attention mechanisms, Transformers can efficiently process both modalities and capture their interactions.

3.3.2.2 Understanding Transformer Model Architecture

The key components of the Transformer model are self-attention mechanisms and feed-forward neural networks, which enable it to capture dependencies between words in a sequence and model complex relationships in the data.

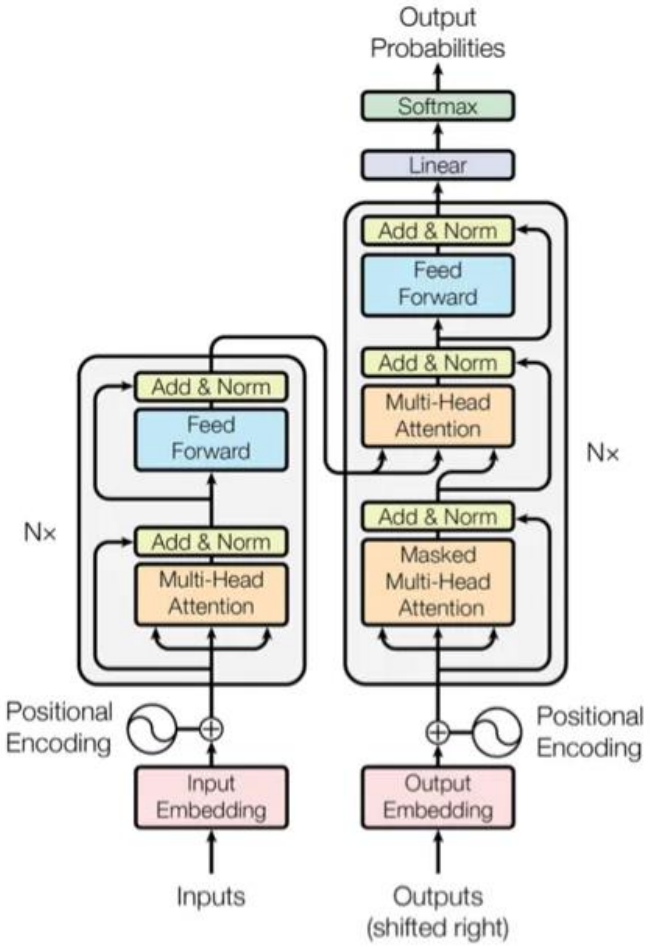


Figure 3.3.2. Transformer Model Architecture

1. Input Embeddings: The input sequence is initially transformed into continuous vector representations called embeddings. These embeddings capture the semantic meaning of each word or token in the sequence.

2. Positional Encoding: Positional encoding is added to the input embeddings to incorporate positional information into the model. This allows the Transformer to understand the sequential order of the words without relying on recurrent connections.

3. Encoder: The encoder consists of a stack of identical layers, each containing two sub-layers: a multi-head self-attention mechanism and a feed-forward neural network. The self-attention mechanism allows the model to attend to different words in the input sequence, capturing their relationships and dependencies. The feed-forward neural network helps to process the attended representations and apply non-linear transformations.

4. Decoder: Similar to the encoder, the decoder also consists of a stack of identical layers. In addition to the self-attention mechanism and feed-forward neural network, the decoder includes an additional multi-head attention mechanism over the encoder's output. This mechanism helps the decoder focus on relevant parts of the input sequence during the decoding process.

5. Masking: To prevent the model from attending to future words during training, the decoder applies a masking technique. This ensures that during the prediction of each word, only the previously generated words are attended to.

6. Final Linear Layer: The output of the decoder is passed through a linear layer followed by a softmax activation to generate a probability distribution over the vocabulary. This distribution represents the predicted word at each position in the output sequence.

Training the Transformer model typically involves optimizing the model parameters using techniques like backpropagation and gradient descent, along with techniques like label smoothing and layer normalization to improve performance and stability.

The Transformer's ability to capture long-range dependencies, parallelize computation, and leverage self-attention mechanisms has made it highly effective for a wide range of NLP tasks, including machine translation, text summarization, question answering, sentiment analysis, and more.

3.3.2.3 Advantages of Transformers

1. Contextual Understanding: Transformers excel at capturing the contextual information and dependencies between words in a text sequence. This ability allows them to generate more coherent and accurate summaries by understanding the relationships between different parts of the text.

2. Attention Mechanism: The attention mechanism in Transformers enables the model to focus on relevant words and phrases when generating the summary. This mechanism helps in

identifying important information and capturing the salient details, resulting in more informative and concise summaries.

3. Handling Variable-Length Input and Output: Transformers can effectively handle variable-length input sequences, which is crucial in text summarization where the length of the source document may vary. They can also generate summaries of varying lengths, accommodating the requirements of different summarization tasks.

4. Transfer Learning: Pretrained Transformer models, such as BERT or T5, can be fine-tuned for text summarization tasks. Transfer learning with pretrained models allows for leveraging large-scale pretraining datasets, improving the performance of the summarization model even with limited labelled data.

3.3.2.4 Disadvantages of Transformers

1. Computationally Intensive: Training and deploying Transformer models for text summarization can be computationally expensive and resource-intensive, requiring significant computational resources and time.

2. Interpretability: Transformers are often considered as black-box models, making it challenging to interpret their decision-making process and understand how they arrived at a particular summary. This lack of interpretability can be a drawback in some applications that require explainable results.

3. Training Data Requirements: Transformers may require a substantial amount of labelled data for training to achieve optimal performance in text summarization. Acquiring and pre-processing such large-scale labelled datasets can be time-consuming and expensive.

4. Handling Rare or Out-of-Vocabulary Words: Transformers can struggle with handling rare words or words not present in the training data, which can lead to issues in generating accurate summaries when encountering such words.

5. Difficulty with Extractive Summarization: Transformers are often used for abstractive text summarization, where they generate summaries in their own words. However, they may face challenges in performing extractive summarization, which involves selecting and combining important sentences or phrases from the source text.

3.3.2.5 Various Transformers used for the task of Summarization

Several Transformer models have been successfully used for abstractive text summarization.

1. BART (Bidirectional and Auto-Regressive Transformers): BART is a Transformer model specifically designed for text generation tasks like abstractive summarization. It incorporates a denoising autoencoder objective during pretraining and has demonstrated strong performance on summarization benchmarks.

2. T5 (Text-To-Text Transfer Transformer): T5 is a versatile Transformer model that can be used for various NLP tasks, including abstractive text summarization. By formulating summarization as a text generation task, T5 has shown impressive performance on summarization benchmarks.

3. Pegasus: Pegasus is a Transformer-based model developed by Google Research for abstractive text summarization. It is trained using a combination of self-supervised objectives, including masked language modelling and document reconstruction. Pegasus has achieved competitive results on summarization tasks.

4. ProphetNet: ProphetNet is a variant of the Transformer model that utilizes a self-supervised objective called sequence-level knowledge distillation. It has been successful in abstractive summarization tasks, particularly in scenarios with limited labelled data.

5. BART Large: BART Large is a larger variant of the BART model that incorporates more parameters. It has shown improved performance on various text generation tasks, including abstractive summarization, although it requires more computational resources.

3.4 Information Retrieval (IR)

In today's data-driven business environment, making accurate decisions relies on extracting relevant information efficiently from large volumes of data. Effectively navigating through this vast sea of information is crucial for identifying key insights that can inform and guide decision-making processes.

This task can be achieved by leveraging NLP algorithms specifically designed for information extraction such as-

1. BERT: BERT is a powerful pre-trained model that has achieved state-of-the-art results in various NLP tasks, including information extraction. It can be fine-tuned for specific tasks such as named entity recognition (NER) to extract entities like organization names, person names, etc., from the text.

2. RoBERTa: RoBERTa is another variant of BERT that further improves the model's performance. It is trained on a larger corpus and with different training techniques. RoBERTa can be effective for information extraction tasks, especially when fine-tuned for specific use cases.

3. GPT: GPT models, such as GPT-2 and GPT-3, are designed for language generation tasks but can also be used for information extraction. By conditioning the model with a query or a prompt, you can leverage its language understanding capabilities to extract relevant information from the public disclosure data.

4. ELMO: ELMO is a deep contextualized word representation model that generates contextualized word embeddings based on the surrounding words in a sentence. It captures the context-dependent meanings of words, making it useful for information extraction tasks.

5. T5: T5 is a versatile pre-trained model that can be fine-tuned for various NLP tasks, including information extraction. It follows a "text-to-text" framework, where different tasks are cast as text generation problems. By framing your information extraction task accordingly, one can leverage T5's capabilities.

3.5 Methodology Used

In the domain of text summarization, various transformer-based models have been developed, each with its own strengths and characteristics as seen in section 3.3.2.5.

3.5.1 Text Summarization

After studying strengths, weakness and characteristics of the models, two models were chosen to perform the task of text summarization- T5 and Pegasus.

3.5.1.1 T5 Model

T5 follows a "text-to-text" framework, where different tasks are cast as text generation problems. By framing the task as a text generation problem, T5 can generate concise summaries given an input text. T5 is pre-trained on a massive corpus, which helps it capture a broad range of language patterns, including those specific to the insurance domain.

T5 employs an encoder-decoder architecture, similar to traditional sequence-to-sequence models, allowing it to generate summaries based on input articles. The encoder encodes the input text, capturing its context and meaning, while the decoder generates the summary based on that encoded representation.

Also, T5 benefits from transfer learning, leveraging the knowledge acquired during pre-training to improve performance on specific downstream tasks like text summarization. By fine-tuning T5 on insurance-specific data, it can further adapt and specialize in summarizing news articles in the insurance domain.

3.5.1.2 Pegasus Model

One of the popular choices for text summarization is Pegasus. The primary reason for selecting Pegasus in this study was its pre-training on a large dataset of news articles. The model is pre-trained on CNN/DailyMail summarization datasets.

The transformer architecture in Pegasus consists of encoder and decoder layers, each containing multiple self-attention (multi-head attention) layers and feed-forward neural networks. The self-attention mechanism allows the model to capture dependencies and relationships between

words or tokens in the input sequence. By attending to different parts of the input sequence, Pegasus can generate informative and coherent summaries.

The transformer architecture used in Pegasus enables it to effectively handle the complexities and dependencies present in sequential data, making it well-suited for text summarization. Through pre-training on a large corpus of news articles, Pegasus learns to extract key information and generate high-quality summaries.

Pegasus was specifically trained using a massive corpus of news articles, which allowed it to capture the nuances and language patterns prevalent in news content. This made it well-suited for summarizing news articles and extracting key information effectively.

The large-scale pre-training of Pegasus on news articles provided it with a strong foundation of language understanding, enabling it to generate high-quality summaries. Fine-tuning Pegasus on a specific domain dataset, insurance in this case, further enhanced its performance and adaptability to the domain-specific need of the study.

3.5.1.3 BART Model

BART is a pre-trained sequence-to-sequence model that can be used for text summarization, among other tasks. It is based on the Transformer architecture and has been trained using a denoising autoencoder objective, which involves corrupting the input text and training the model to reconstruct the original text.

When it comes to text summarization of news articles, BART can be an effective choice due to its ability to generate coherent and fluent summaries. BART can be used by implementing following approaches-

1. Input Encoding: Tokenize the input news article using the BART tokenizer, which splits the text into sub word units. These tokens are then encoded into numerical representations that BART can understand.

2. Model Architecture: BART consists of an encoder-decoder architecture, where the encoder processes the input article and the decoder generates the summary. The encoder and decoder are composed of several layers of self-attention and feed-forward neural networks.

3. Training: BART is pre-trained on large-scale datasets using denoising autoencoder objectives, which involve corrupting the input text by randomly masking or shuffling words

and training the model to reconstruct the original text. This pre-training helps BART learn useful representations of the input text.

4. Decoding: During inference, given a new article, BART generates the summary by autoregressively decoding tokens one by one. At each step, the model predicts the most likely token based on the previous tokens generated.

Fine-tuning BART on a large dataset of paired news articles and summaries is recommended to achieve better performance on specific summarization tasks.

3.5.2 Information Retriever

Section 3.4 presented a compilation of pre-trained models suitable for information retrieval based on queries. From this, the study selected two models, namely BERT and RoBERTa, for implementation in the research. These models were chosen considering their capabilities, applicability to the study's objectives, and their prominence in the field.

3.5.2.1 BERT Model

When it comes to information retrieval, BERT can be used to retrieve relevant documents or passages based on a given query by implementing following approach:

1. Pre-processing: Tokenizing the query and the documents into BERT-compatible tokens. BERT uses Word Piece tokenization, which splits words into sub word units. Additionally, truncate or pad the tokens to a fixed length to fit the model's input requirements.

2. Encoding: Converting the tokenized query and document passages into their corresponding numerical representations using BERT. This process involves mapping each token to its unique ID and generating the token type and attention masks.

3. Ranking: Calculating the similarity between the encoded query and each document passage. One common technique is to use cosine similarity, where the higher the cosine similarity score, the more similar the query and document are. Calculating the similarity by taking the dot product between the encoded query and document vectors and normalizing them.

4. Retrieval: Sorting the document passages based on their similarity scores in descending order to retrieve the most relevant documents or passages for the given query.

Implementing information retrieval with BERT requires pre-trained BERT models, such as "bert-base-uncased" or "bert-large-uncased," as well as a suitable training or fine-tuning process on specific dataset to adapt BERT to the retrieval task at hand.

3.5.2.2 RoBERTa Model

The approach employed with the RoBERTa model shares many similarities with the BERT model. RoBERTa (Robustly Optimized BERT Pretraining Approach) is an extension of the BERT (Bidirectional Encoder Representations from Transformers) model, and it improves upon several aspects of BERT, making it potentially better suited for information retrieval tasks.

Some key reasons why RoBERTa can be advantageous:

1. Larger and Longer Pretraining: RoBERTa is trained on a much larger corpus of 160GB compared to BERT's 16GB. This extended training duration allows RoBERTa to capture a broader range of language patterns and semantic nuances.

2. Dynamic Masking during Pretraining: RoBERTa employs dynamic masking, randomly selecting masked token positions for each training epoch. This approach enhances the model's ability to understand context by reducing reliance on positional information.

3. Removed Next Sentence Prediction (NSP) Objective: Unlike BERT, RoBERTa removes the NSP objective, focusing more on sentence-level understanding and capturing relationships within and across sentences.

4. Training Hyperparameter Optimization: RoBERTa extensively explores hyperparameter configurations and training setups to optimize performance. It investigates variations in batch size, training duration, dynamic masking, and other parameters.

3.6 Evaluation Approaches

Two approaches were utilized for the analysis: manual evaluation and metric-based assessment.

3.6.1 Manual Evaluation

The manual analysis involved the following aspects when examining the generated summaries:

1. Coherence: The coherence of the summary was assessed by evaluating the logical and smooth flow of sentences.

2. Informativeness: The extent to which the summary captured the key points and crucial information from the original text was examined.

3. Conciseness: The length of the summary was considered, with emphasis on conveying the main points concisely using minimal text.

4. Grammar and Fluency: The grammatical correctness and overall fluency of the generated text were reviewed.

5. Contextual Understanding: The summary was evaluated for its ability to demonstrate a comprehensive understanding of the context and nuances of the original text, ensuring accurate representation without misinterpretations.

6. Comparative Analysis: Summaries generated by different models were compared to identify variations in quality, content, and style. This comparative analysis aided in recognizing the strengths and weaknesses associated with each approach.

3.6.2 Metric-based Evaluation

There are several evaluation metrics used for text summarization evaluation. Here are some commonly employed metrics:

1. ROUGE (Recall-Oriented Understudy for Gisting Evaluation): ROUGE is a set of metrics that measure the overlap between the generated summary and the reference summaries. It includes metrics like ROUGE-N (n-gram overlap), ROUGE-L (longest common subsequence), ROUGE-S (skip-bigram overlap), and ROUGE-SU (skip-bigram and unigram overlap). ROUGE metrics focus on recall and assess the quality of summaries based on content overlap.

2. BLEU (Bilingual Evaluation Understudy): BLEU measures the similarity between the generated summary and the reference summaries using n-gram precision. It compares the n-grams in the generated summary against the reference summaries and calculates a score based on their overlap. BLEU is commonly used in machine translation evaluation but can also be applied to text summarization.

3. METEOR (Metric for Evaluation of Translation with Explicit ORdering): METEOR is an evaluation metric that combines precision, recall, and alignment-based matching. It considers exact word matching, stemmed matching, and paraphrase matching to compute a score. METEOR incorporates linguistic and semantic information to evaluate summary quality.

4. CIDEr (Consensus-based Image Description Evaluation): While initially designed for image captioning, CIDEr has been adapted for text summarization evaluation. It measures the consensus between the generated summary and the reference summaries based on n-gram similarity. CIDEr takes into account not only content overlap but also diversity in the generated summaries.

5. F1 Score: F1 score is a common metric that combines precision and recall. In text summarization, F1 score can be used to evaluate the overlap between the generated summary and the reference summaries at various levels, such as unigram, bigram, or sentence.

The choice of the "best" evaluation metric for text summarization depends on several factors and the specific requirements of the task at hand. However, considering the nature of insurance news articles, ROGUE seemed to be particularly relevant as ROUGE metrics are widely used and suitable for evaluating content overlap between the generated summary and the reference summaries. Since insurance news articles often contain important information and specific details, ROUGE metrics can assess how well the generated summary captures the key points and facts mentioned in the references.

3.6.2.1 ROGUE Metrics

The ROUGE metrics are commonly used for evaluating the quality of automatic summaries compared to reference summaries. ROUGE evaluates the overlap between the generated summary and the reference summaries by calculating various recall-based measures. These metrics assess the quality of summaries based on the matching of n-grams, word sequences, and sentence-level information.

1. ROUGE-N: ROUGE-N measures the n-gram overlap between the generated summary and the reference summaries. The "N" refers to the length of the n-grams being considered. For example, ROUGE-1 measures the unigram overlap, ROUGE-2 measures the bigram overlap, and so on. ROUGE-N calculates the recall of n-grams, capturing the extent to which the generated summary contains similar word sequences as the references.

$$ROUGE - n = \frac{\sum_{S \epsilon RS} \sum_{gram_n \epsilon S} Count_{match}(gram_n)}{\sum_{S \epsilon RS} \sum_{gram_n \epsilon S} Count(gram_n)}$$

- reference summary is a set denoted by RS
- length of $n-gram$ is denoted by n
- maximum number of $n-grams$ co-occurring in the output summary for a set of references is denoted by *Count match (gram n)*

2. ROUGE-L: ROUGE-L computes the longest common subsequence (LCS) between the generated summary and the reference summaries. It measures the recall of the LCS, which represents the longest contiguous sequence of words shared by both the generated and reference summaries. ROUGE-L is useful in evaluating summaries that rephrase sentences but still maintain the core content.

3. ROUGE-S: ROUGE-S calculates the skip-bigram overlap between the generated and reference summaries. Skip-bigrams are pairs of words that are not necessarily adjacent but maintain the order of occurrence in the sentence. ROUGE-S captures the recall of skip-bigrams, which helps evaluate summaries that rearrange words while preserving the overall sentence.

ROUGE metrics provide different perspectives on the quality of summaries, taking into account different levels of matching, such as individual words (ROUGE-N), sentence structures (ROUGE-L), and skip-bigrams (ROUGE-S). The scores for these metrics range from 0 to 1, where a higher score indicates a better match between the generated and reference summaries.

While ROUGE metrics are widely used, they have some limitations. They rely solely on lexical overlap and do not capture semantic or syntactic information. Additionally, ROUGE metrics do not consider the coherence, readability, or overall meaning of the summaries, which can be essential aspects of summary evaluation. Therefore, it's recommended to use ROUGE metrics as one part of a comprehensive evaluation approach, considering other factors and human judgment for a more holistic assessment of summary quality.

3.7 Proposed Method

The overall methodology for this research work is outlined in the following Research workflow.

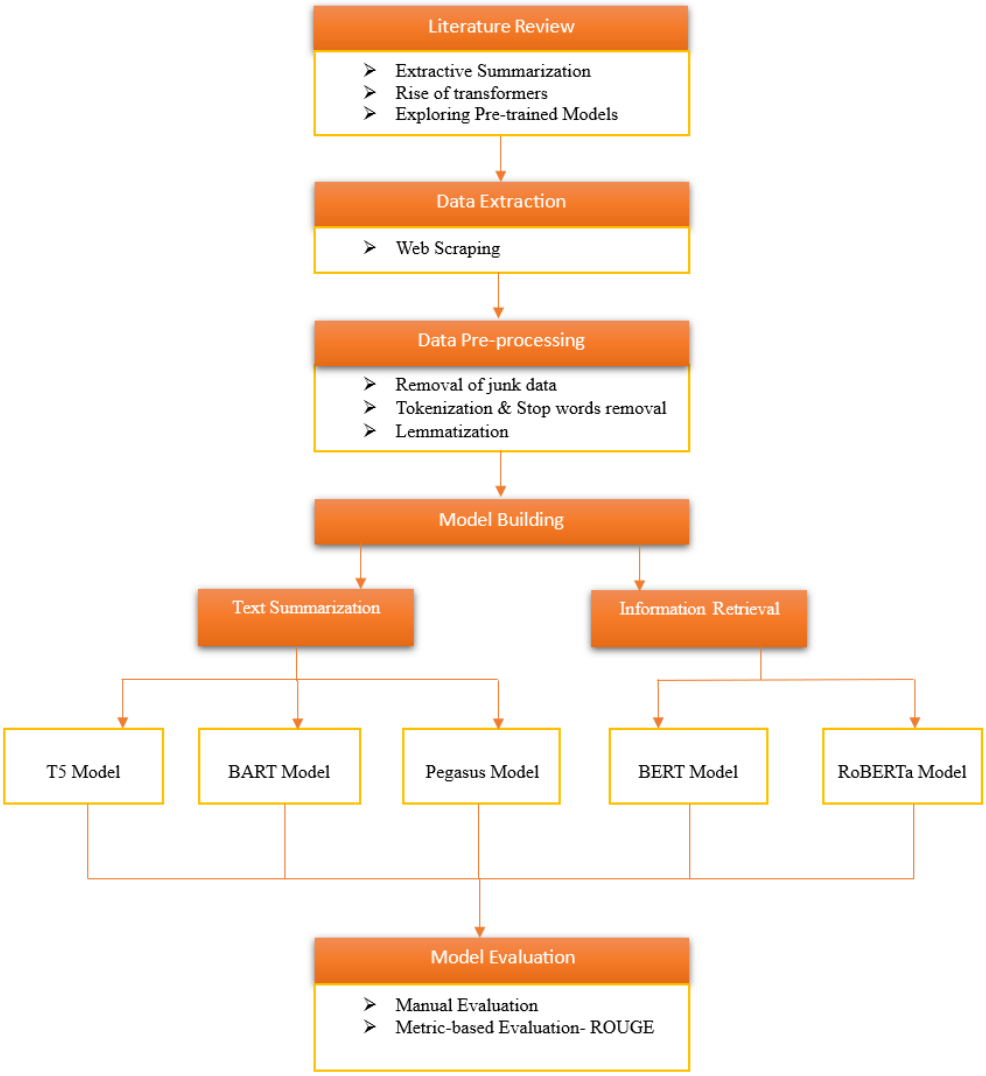


Figure 3.7 Research Workflow

1. Literature Review: The research begins with an extensive review of existing studies in the field of Text Summarization and Information Extraction using Natural Language Processing (NLP). This step aims to gain a thorough understanding of the current state of research in these areas and identify relevant techniques and approaches.

2. Data Extraction: In this step, web scraping techniques are employed to extract news articles from various online sources. Additionally, public disclosure files are downloaded to augment the dataset. The collected data will serve as the basis for the subsequent analysis and model training.

3. Data Pre-processing: Before feeding the data into the models, several pre-processing steps are performed. These include data cleaning to remove any irrelevant or noisy information, tokenization to break the text into smaller units (e.g., words or phrases), and removal of stop words that do not contribute significantly to the overall meaning. These pre-processing steps help prepare the data for the subsequent modelling stages.

4. Model Selection: Three pre-trained models are selected for text summarization, and two models are chosen for information extraction. The selection is based on their performance in relevant tasks and their suitability for the research objectives. These pre-trained models provide a foundation for building accurate and effective summarization and extraction systems.

5. Model Evaluation: To assess the performance and effectiveness of the selected models, two evaluation methods are employed. First, manual analysis is conducted by human annotators who assess the quality of the generated summaries and extracted information. This qualitative analysis provides valuable insights into the strengths and weaknesses of the models. Second, metric-based evaluation is performed, where established evaluation metrics specific to text summarization and information extraction tasks are used to quantitatively measure the models' performance.

The research is being carried out on personal laptop, following are system configuration and software requirements.

Table 3.7 System requirements

Hardware Configurations	**Software Requirements**
Processor: AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz	OS: Windows 10
System type: 64-bit operating system, x64-based processor	Python version: 3.10
Wireless Networking: 802.11n	Code Editor: Visual Studio Code/ Anaconda/ JupyterNB
RAM: 24.0 GB	Libraries: Pandas, Numpy, Sklearn, NLTK, TensorFlow, Keras, transformers etc
GPU: Nvidia GeForce RTX 3060 6GB	Other Software: Microsoft Office (Excel, Word, Power Point, One Note), Google Chrome, Adobe Acrobat Reader, Notepad, 7-Zip, Tableau etc

3.8 Summary

The study emphasized the importance of reliable and relevant data for competitive intelligence. News articles and public disclosures were chosen as primary data sources. Manual data collection through web scraping was proposed, and text summarization strategies using Seq2Seq and Transformer models were discussed for effective processing of sequential information. In the domain of text summarization, various transformer-based models were developed and studied. The T5 and Pegasus models were chosen for text summarization. T5 followed a "text-to-text" framework, while Pegasus was pre-trained on a large dataset of news articles. The BART model, based on the Transformer architecture, was also discussed. The evaluation involved manual assessment and metric-based evaluation using metrics such as ROUGE, BLEU, METEOR, CIDEr, and F1 score. ROUGE metrics would be particularly relevant for summary evaluation.

CHAPTER 4

ANALYSIS AND IMPLEMENTATION

4.1 Introduction

This chapter focuses on data analysis and implementation of various models discussed in previous chapter.

4.2 Data Extraction

The data extraction process involved utilizing the Google SERP API to search for news articles related to specific competitors. This was achieved by using the competitors' names as query keywords, allowing to extract the relevant news article links associated with those queries. Reframe: The data extracted from the obtained news article links was collected through the process of web scraping. This involved utilizing Python as the programming language and several libraries such as Pandas, serpapi, beautifulsoup, and trafilatura.

Python served as the programming language of choice, providing a robust and versatile environment for web scraping tasks. The Pandas library was utilized for efficient data handling and manipulation, allowing for easy organization and analysis of the scraped data.

To facilitate the web scraping process, the serpapi library was employed to interact with the Google SERP API and retrieve the HTML content of the news articles associated with the given query keywords. This allowed for the automated retrieval of the desired information from the search engine results.

The beautifulsoup library played a crucial role in parsing and extracting the relevant data from the HTML content. It provided powerful tools for navigating and extracting specific elements from the web pages, enabling the extraction of valuable information such as article titles, publication dates, authors, and article texts.

In addition, the trafilatura library was used for text extraction from HTML, ensuring the retrieval of clean and readable text content from the news articles.

The data was stored in csv file format as CSV files are preferred over Excel in pandas due to their simplicity, lightweight nature, compatibility with various tools, faster processing speed, and better compatibility with version control systems.

To gather public disclosure files, the process involved visiting the official websites of competitors. Upon navigating through the websites, “About us” section was located, which typically contained various options such as Press releases, public disclosures, Media Centre, and more. Within one of these options, the files related to public disclosures were found, conveniently organized by quarter. The files were downloaded from the website’s interface, allowing for easy access and retrieval of the necessary documents for further analysis and study.

4.3 Data Description

Through the process of web scraping, around 1500 news articles were collected from prominent insurance companies including HDFC Life, ICICI Prudential, LIC, Tata AIA, and Max Life. The data obtained from the web scraping exercise was then structured and stored in six columns in a CSV file. This approach ensured convenient storage, organization, and analysis of the news article dataset.

Table 4.3. Dataset with column description

Sr No.	Column Name	Column Description
1	Title	Headline of the News Article
2	Links	Link of the Article
3	Date	Date on which the article is published
4	Source	News channel name
5	Article_1	Content of the article scraped using web scraping method 1
6	Article_2	Content of the article scraped using web scraping method 2

During the scraping process, two different methods were employed to extract articles: one using the Beautiful Soup library and the other using the Trafilatura library. The aim was to ensure high-quality and relevant content. A comparison was made between the articles obtained

through both methods, considering factors such as article length, the presence of junk content, and excessive white spaces. Based on this evaluation, content generated by the Beautiful Soup library was preferred for certain articles, while content generated by Trafilatura was chosen for others. This content was stored in a new column created in the dataset to specifically store the article content intended for training.

In addition to the content, other information such as the article's headline, the source link from which it was obtained, the publication date, and the article's source were also extracted. However, these additional data points were not utilized during the model training phase. Instead, they served as valuable references for analysis and further investigation.

4.4 Data Preparation

The availability of data from standard data websites was limited as the problem statement of the study was domain-specific, focusing on a particular industry or niche. As a result, the required data could not be obtained directly from existing data repositories or readily accessible sources.

To address this challenge, an extensive effort was made to extract the necessary data from various sources relevant to the domain. These sources included industry-specific websites, research publications, public disclosures, news articles, and other relevant platforms.

However, the collected data was not in a structured format suitable for immediate analysis. It was cluttered, disorganized, and contained noise and inconsistencies due to its varied origins. To ensure the data's quality and reliability, a rigorous data cleaning process was undertaken.

Following are the data cleaning steps that were performed on the news articles data:

4.4.1 HTML tags Removal

Since the data was extracted through scraping, it is common to encounter HTML tags in the extracted text. To clean the data and prepare it for further analysis or modelling, the first step was to remove these HTML tags.

HTML tags contain formatting and structural information specific to web pages but are not relevant to the textual content itself. Removing these tags helped to extract the meaningful text and ensured consistency in the dataset.

By removing HTML tags, the extracted data became more readable, consistent, and suitable for summarization task. This step was useful in order to focus on the actual content of the articles rather than the underlying web page structure.

4.4.2 Duplicate Article Removal

During the scraping process, it was observed that some articles appeared multiple times in the dataset. To ensure unbiased and accurate model performance, it was crucial to remove these duplicate articles. If such articles were to remain in the dataset, they could introduce a bias and potentially skew the model's training and evaluation.

Removing duplicate articles helped to maintain the integrity and fairness of the dataset, allowing the model to learn from a diverse and representative set of articles. This step contributed to reducing any potential biases and ensuring the model's performance was not compromised by redundant information.

4.4.3. Special characters and Whitespaces elimination

To enhance the quality of the text for the summarization process, special characters like punctuation marks, whitespace, and non-printable characters were removed. These characters can introduce noise and unnecessary variations in the text, which may affect the accuracy and effectiveness of the summarization algorithm.

By eliminating special characters, the text became more streamlined and focused on the essential content.

4.4.4. Lowercasing and Spell check

To address the issue of duplication caused by words with different capitalizations, the text was converted to lowercase. By applying this transformation, all words in the text were changed to their lowercase form, ensuring uniformity in the representation of words.

This conversion helps to eliminate the distinction between words with different capitalizations, such as "Hello" and "hello," treating them as the same word. This step is particularly important in text processing tasks like summarization, where the focus is on the semantic meaning of words rather than their case sensitivity.

To ensure the accuracy and reliability of the text data, spell-checking libraries were utilized to handle spelling errors. By leveraging spell-checking libraries, such as PySpellChecker, the extracted text was subjected to automated spell-checking. The library utilizes dictionaries and language models to identify and correct misspelled words within the text.

During the spell-checking process, each word in the text was compared against a dictionary of correctly spelled words. If a word was found to be misspelled, suggestions for correct replacements were provided. The misspelled words were then replaced with the suggested corrections to rectify the errors.

4.4.5 Tokenization

Tokenization is a crucial step in text analysis as it breaks down the text into individual words or tokens, allowing for a more granular analysis of the text's structure and content. In the case of the extracted articles, tokenization was performed using libraries like NLTK.

By tokenizing the text, each word or token was isolated, creating a structured representation of the content. This process enabled subsequent analysis at the word level, such as counting word frequencies, identifying key phrases, or applying machine learning algorithms.

By using libraries like NLTK, which provide robust tokenization capabilities, the extracted articles were effectively transformed into a sequence of tokens, facilitating further analysis and processing to support text summarization.

4.4.6 Stop word Removal

Stop words, which are commonly used words that do not contribute significantly to the overall meaning of a text, can introduce noise and unnecessary complexity in the summarization process. Examples of stop words include articles (e.g., "a," "an," "the"), prepositions (e.g., "in," "on," "at"), and pronouns (e.g., "he," "she," "it").

To improve the quality of summarization and reduce noise, stop words were removed from the text. Libraries like NLTK provide pre-defined lists of stop words for different languages, including English. These lists contain commonly occurring words that are considered as stop words.

By removing stop words from the text, the focus was shifted to the more meaningful and informative words, enabling a clearer understanding of the content. This step helped in improving the efficiency and effectiveness of the summarization process, as the resulting summary would be more concise and relevant, containing key information without unnecessary clutter.

4.4.7 Irrelevant content Removal

To ensure that only relevant information is considered during the summarization process, non-relevant content such as advertisements, comments, or other unrelated information present in the news articles was excluded. This step was crucial for maintaining the focus on the main text of the articles and generating accurate and concise summaries.

By excluding non-relevant content, the summarization algorithm can concentrate on the actual article content that provides valuable information and context. This improves the quality of the summaries by eliminating noise and distractions that may hinder the understanding and extraction of key points.

4.4.8 Normalization

As part of the text normalization process, several steps were taken to enhance the quality and consistency of the text. These steps included removing multiple spaces, handling abbreviations, and expanding contractions.

1. Removing multiple spaces: Extra spaces between words were eliminated to ensure consistent spacing and formatting throughout the text. This helped improve readability and ensured accurate tokenization.

2. Handling abbreviations: Abbreviations present in the text were expanded to their full forms to enhance comprehension and avoid ambiguity.

3. Expanding contractions: Contractions, such as "don't" or "can't," were expanded to their full forms. This step helped maintain consistency and clarity in the text, as contractions can sometimes be ambiguous or unclear in meaning.

4.4.9 Lemmatization

Lemmatization was conducted on the article's text which contributed to improved semantic understanding and the generation of high-quality summaries.

By reducing words to their base or dictionary form (lemmas), Lemmatization helps in capturing the core meaning of the words while maintaining grammatical correctness. Compared to stemming, which simply removes prefixes or suffixes to obtain the base form, lemmatization takes into account the context and part of speech of the word. This allows for more precise normalization of words and helps in generating more meaningful summaries.

Clean and well-prepared data ensured that the model received accurate and reliable information during training. By removing inconsistencies, errors, and noise from the data, I was able to reduce the chances of the model being misled or biased by incorrect or irrelevant information. This helped improve the overall accuracy and performance of the trained model.

4.5 Model Implementation

4.5.1 Summarization Using Pegasus Model

The implementation consisted of a function designed to perform text summarization using the Pegasus model. The function took an input text and a maximum word count as parameters, and initialized the Pegasus model and tokenizer using the pre-trained google/pegasus-xsum model.

To facilitate the summarization process, the input text is split into smaller chunks based on the specified maximum word count. This step ensures that the text is processed in manageable portions to generate accurate and concise summaries. Next, the function iterates through each chunk of text. For each chunk, the Pegasus model generates a summary using the tokenizer. The summary is decoded to remove special tokens and unnecessary spacing. To meet the desired maximum word count, the summary is trimmed by selecting only the first N words, where N is

the specified maximum word count. This step ensures that the generated summaries are concise and within the specified length limit.

4.5.2 Summarization using T5 Model

In this a tokenizer was created using the T5 Tokenizer class from the t5-base pre-trained model. A model was initialized using the T5ForConditionalGeneration class from the t5-base pre-trained model. The encoded inputs were passed to the model's generate method, which generated the summary based on the inputs. The generated summary had a maximum length of 250 tokens and used 4 beams for beam search decoding. Early stopping was enabled to stop generating further tokens when the model predicted the end of the text. The generated summary was decoded using the tokenizer to convert it into readable text. Finally, the summary was returned as the output of the summarize_article function.

4.5.3 Summarization using BART Model

Similar to T5 pre-trained model, a tokenizer was created using the BART Tokenizer class, which was loaded from the facebook/bart-large-cnn pre-trained model. Then, a model was initialized using the BARTForConditionalGeneration class, also from the facebook/bart-large-cnn pre-trained model. To generate the summary, the inputs were encoded using the tokenizer. These encoded inputs were then passed to the 'generate' method of the model, which employed beam search decoding with 4 beams to generate the summary. The maximum length of the generated summary was set to 250 tokens, and early stopping was enabled to stop generating additional tokens once the model predicted the end of the text.

The generated summary was obtained by decoding the model's outputs using the tokenizer, converting it into readable text. The output of the summarize_article function was the summary which was generated.

4.5.4 Information Retrieval using BERT Model

A solution for extracting information from tables in a PDF document using the BERT model was implemented. Necessary libraries including torch, spacy, fitz, tabula, BertTokenizer, and BertForQuestionAnswering were imported. Tabula library was used to extract tabular data from

pdf file. The PDF file was loaded using the fitz library which extracted the text that was further passed through spaCy's natural language processing pipeline for further processing and analysis. The BERT model and tokenizer were employed and the query and table text were encoded using the BERT tokenizers encode_plus method, with a maximum sequence length of 512. The encoded inputs were then passed to the BERT model, which generated start and end logits. The indices with the highest logits are used to extract the answer from the table.

4.5.5 Information Retrieval using RoBERTa Model

Necessary libraries, including torch, RobertaTokenizer, RobertaModel, and PyPDF2 were imported. The pre-trained RoBERTa model and tokenizer were loaded. A function named extract_text_from_pdf was defined to extract text from the PDF file using the PyPDF2 library. The PDF text was then split into individual disclosures using the newline separator. The query was tokenized using the tokenizer, and a function named encode_segments was defined to encode text segments using the tokenizer. The disclosure data was then encoded in smaller segments, with a specified segment size. The input IDs and attention masks for each segment were stored. The segmented input IDs and attention masks were concatenated into a single tensor. The model was used to obtain embeddings for the disclosure data by passing the concatenated input IDs and attention masks. Similarity scores between the query and each disclosure were calculated using cosine similarity. The most relevant disclosure was determined by finding the index with the highest similarity score.

4.6 Summary

The data extraction process involved using the Google SERP API to search for news articles related to specific competitors. Python and libraries such as Pandas, serpapi, beautifulsoup, and trafilatura were used for data extraction and web scraping. Around 1500 news articles were collected, structured, and stored in a CSV file with columns for title, links, date, source, and two article content columns. Data preparation steps included removing HTML tags, duplicate articles, irrelevant content, Tokenization, stop word removal, normalization and lemmatization were also performed. The data was then used for text summarization using Pegasus, T5, and BART models, pdf files were used for information retrieval using BERT and RoBERTa models.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction

This chapter presents the outcomes of the study, which involved employing pre-trained models such as T5 and Pegasus for text summarization of extracted, cleaned, and processed news articles. Additionally, key information was obtained from public disclosure data using BERT and RoBERTa models. The results are thoroughly examined, compared, and evaluated in this chapter.

5.2 Evaluation of Text summarization models

To summarize news articles, three pre-trained models, namely T5, BART, and Pegasus, were employed. The subsequent step involved analysing the generated summaries and comparing them with summaries produced by other models.

To facilitate evaluation and discussion, we selected a reference summary generated manually as an example. All the generated summaries were compared against reference summary and their similarity was quantified using ROUGE scores.

Reference Summary:

HDFC Life has introduced Click2Protect Super, a term insurance plan that offers customization based on individual protection needs. The plan is a non-linked, non-participating, individual life insurance policy that provides flexibility in terms of changing life cover, extending the policy term, and obtaining coverage for accidental death and terminal illness. Click2Protect Super offers three plan options: Life, Life Plus, and Life Goal, each with its own set of benefits. The plan aims to provide comprehensive financial protection to families and allows policyholders to choose the coverage amount and customize the plan according to their requirements. The product reflects HDFC Life's commitment to meeting the evolving needs of Indian consumers and offering flexible and customer-centric insurance solutions.

The table 5.2 demonstrates results obtained from different model which have been compared with the above-mentioned reference summary.

Table 5.2.1 Comparison of Summaries generated by models

Models	Summaries generated by the model
T5	Click2Protect Super is a non-linked, non-participating, individual, pure risk premium/ savings life insurance plan. it provides comprehensive -financial protection for the family. one can choose a cover as per requirement from among three plan options - Life, Life Plus & Life Goal
BART	HDFC Life launches Click2Protect Super term insurance policy. The plan provides comprehensive -financial protection for the family. One can choose a cover as per requirement from among three plan options – Life, Life Plus & Life Goal. Click 2 Protect Super will be the new age financial safety net for customers
Pegasus	HDFC Life has launched a new term insurance plan Click2Protect Super, which enables customisation as per protection needs, the company said in a statement. Click2Protect Super is an individual, pure risk premium/ savings life insurance plan that provides you with flexibilities such as changing life cover, extending the policy term and terminal illness among others. One can choose a cover as per requirement from among three plan options – Life, Life Plus & Life Goal. HDFC Life has launched a new term insurance product, Click2Protect Super, in the Indian market. Click 2 Protect Super is the new age financial safety net for customers who believe in living a life of pride.

Upon analysing the generated summaries by the pre-trained models, it is evident that the summary generated by the T5 model is concise, coherent, grammatically correct, and informative. The BART model has also produced a satisfactory summary that meets all the aforementioned criteria. However, it appears that the BART model missed capturing some information that was successfully captured by the T5 model. On the other hand, the summary generated by the Pegasus model is the longest among the three models. While it demonstrates

contextual understanding, it lacks conciseness. Nevertheless, it does provide additional information not found in the other two summaries.

Table 5.2.2 Evaluation and Comparison of ROUGE Scores

Models	**Evaluation Metrics**		
	ROUGE-1	ROUGE-2	ROUGE-L
T5	0.40	0.33	0.39
BART	0.38	0.28	0.37
Pegasus	0.41	0.29	0.40

While the generated summaries by all the models were deemed satisfactory, a notable observation was made regarding their limited comprehension of domain-specific terminologies. For instance, the model misinterpreted “VoNB” as “Vijaya National Bank” instead of its actual meaning, which is “Value of New Business”. Further evaluation revealed some more instances where such terminologies were inaccurately interpreted, leading to sentence meanings being changed or conveying false information.

Through extensive research, it was revealed that the pre-trained models utilized in the study were trained on vast amounts of generic data, which could lead to potential misinterpretation of certain terminologies. However, this issue can be effectively addressed by training the model on domain-specific data, specifically articles-summary pairs that are representative of the insurance industry in this particular case.

In an attempt to address this challenge, the decision was made to train the Pegasus model using an insurance dataset consisting of article-summary pairs. However, due to limitations in computational power and resources, the training process could not be successfully executed.

5.3 Evaluation of Information Retrieval system

Unfortunately, the performance of the BERT and RoBERTa models fell short of expectations. Initially, these models were producing nonsensical outputs, and attempts to rectify the issue resulted in receiving no relevant information at all. Consequently, despite the attempts to fix and optimize the models, the desired information could not be captured for the given query.

5.4 Summary

Three pre-trained models, T5, BART, and Pegasus, were used to summarize news articles. The generated summaries were compared to a manually generated reference summary using ROUGE scores. The T5 and BART models produced concise and coherent summaries, while the Pegasus model was longer but provided additional information. In terms of ROUGE scores, T5 and Pegasus performed better than BART. However, all models had difficulty comprehending domain-specific terminologies. But due to limitations in computational power and resources, the training process could not be successfully executed. Training the models on domain-specific data could address this issue. The BERT and RoBERTa models performed poorly in information retrieval, failing to capture relevant information despite attempts to optimize them.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

In this chapter, we present the concluding remarks of the study, which focused on the effectiveness of the proposed system utilizing two tools: Text Summarization and Information Retrieval. The chapter provides an overview of the study, highlighting the utilization of these tools for efficient competition benchmarking. A discussion on the outcomes of the research will be done to draw definitive conclusions. Additionally, recommendations for further improvements and enhancements to the system will be provided.

6.2 Discussion and Conclusion

This study revealed that in order for any company to maintain a competitive edge and drive business success, the acquisition of relevant market information is imperative. While conventional approaches to collecting competitive intelligence often involve laborious and time-consuming methods, this study aimed to identify more efficient alternatives.

The research endeavour focused on discovering improved ways to construct a system that not only facilitates the achievement of the aforementioned goals but does so with heightened efficiency. Through this study, practical solutions have been unveiled, providing a pathway to overcome the challenges associated with manual and cumbersome approaches.

To ensure the highest level of performance and efficiency, a comprehensive set of five pre-trained models was employed in this study. Specifically, T5, BART and Pegasus were utilized for text summarization, while BERT and RoBERTa were utilized for Information Retrieval purposes. The integration of these advanced models enabled the extraction of information with remarkable accuracy, unparalleled speed and remarkable capabilities to process and analyse large volumes of data swiftly and effectively.

The proposed system presents significant potential for improvement, particularly through the utilization of domain-specific models trained on specific data. The integration of such models has the potential to yield the most accurate summaries or information within a particular

domain. However, due to limited computational power and resources during the study, training a domain-specific model with a large number of article-summary pairs posed a significant challenge. Given the constraints at the time, the study resorted to leveraging pre-trained models, which required less computational power.

Although the results obtained from the pre-trained models were satisfactory, it is important to acknowledge that they serve as a promising indicator of the proposed system's potential. While they may not fully exploit domain-specific nuances, the pre-trained models still demonstrate the system's capability to establish streamlined methods for gathering competitive intelligence.

As computational power and resources continue to advance, future iterations of the proposed system can explore training domain-specific models, enabling more precise and tailored outcomes. These improvements would enhance the accuracy and effectiveness of the system, further optimizing the gathering of competitive intelligence.

The research sought to identify improved strategies and methodologies to enhance the effectiveness of gathering competitive intelligence. By implementing the methods proposed in this study, companies can benefit from a system that enables them to gain a competitive edge more efficiently. The research aimed to provide practical solutions that overcome the challenges associated with manual approaches, offering streamlined and optimized methods for collecting essential competitive intelligence.

It was indeed disappointing to observe that the Information Retrieval tool encountered difficulties in extracting information from the public disclosure files. The primary challenge that emerged was the data format itself. While the model managed to extract some basic data, it struggled when confronted with complex queries and particularly with data presented in a tabular structure.

The tabular format posed a higher level of complexity for the model, hindering its ability to accurately extract the desired information. This limitation suggests that the model requires further development and refinement to effectively handle such intricate data structures. Or find some other models/architecture that can be used to handle tabular data and extract information out of it.

6.3 Future Recommendations

Recognizing the discussed setbacks in previous section, it became apparent that future iterations of the system need to focus on enhancing the Information Retrieval tool's capabilities to handle diverse data formats, including tabular structures. By addressing this challenge, the system can become more robust and offer a comprehensive solution for extracting information from a wide range of data sources.

Furthermore, the utilization of enhanced computational power and high-end resources presents an opportunity to attain more precise summaries through the application of transfer learning techniques. By leveraging domain-specific data during the training process, models can be fine-tuned to gain more precision in the generated summaries.

It is important to acknowledge the limitations and challenges faced during the study, as they provide valuable insights for further research and improvement. By understanding these hurdles, researchers can work towards developing more sophisticated models and algorithms that can effectively extract information from complex data formats, gain more accurate summaries which will ultimately enhance the performance and reliability of the proposed system.

REFERENCES

Balaji, N., Megha, N., Kumari, D., Kumar, S. and Bhavatarini, N., 2022, October. Text Summarization using NLP Technique. In 2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER) (pp. 30-35). IEEE.

Gupta, A., Chugh, D. and Katarya, R., 2022. Automated news summarization using transformers. In Sustainable Advanced Computing: Select Proceedings of ICSAC 2021 (pp. 249-259). Singapore: Springer Singapore.

Christian, H., Agus, M.P. and Suhartono, D., 2016. Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). ComTech: Computer, Mathematics and Engineering Applications, 7(4), pp.285-294.

Graves, A., 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Nallapati R, Xiang B, Zhou B (2016) Sequence-to-Sequence RNNs for Text Summarization. CoRR abs/1602.06023:

Luhn, H.P., 1958. The automatic creation of literature abstracts. IBM Journal of research and development, 2(2), pp.159-165.

Morris, J. and Hirst, G., 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational linguistics, 17(1), pp.21-48.

Barzilay, R. and Elhadad, M., 1999. Using Lexical Chains for Text Summarization In: Mani I and Maybury MT, eds. Advances in Automatic Text Summarization.

Silber, H.G. and McCoy, K.F., 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. Computational Linguistics, 28(4), pp.487-496.

Choudhary, S., Guttikonda, H., Chowdhury, D.R. and Learmonth, G.P., 2020, April. Document retrieval using deep learning. In 2020 Systems and Information Engineering Design Symposium (SIEDS) (pp. 1-6). IEEE.

Chatterjee, N., Mittal, A. and Goyal, S., 2012, November. Single document extractive text summarization using genetic algorithms. In 2012 Third International Conference on Emerging Applications of Information Technology (pp. 19-23). IEEE.

Patel, S.M., Dabhi, V.K. and Prajapati, H.B., 2017. Extractive Based Automatic Text Summarization. J. Comput., 12(6), pp.550-563.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

Nallapati, R., Zhou, B., Gulcehre, C. and Xiang, B., 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023.

Ashish, V., 2017. Attention is all you need. Advances in neural information processing systems, 30, p.I.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Kuhanec, T. (2022) What Is Competitive Intelligence?. Available at: https://www.meltwater.com/en/blog/competitive-intelligence.

Sprinklr Team (2021) What is Competitive Benchmarking? Definition & Guide (no date). Available at: https://www.sprinklr.com/blog/what-is-competitive-benchmarking/.

APPENDIX A: RESEARCH PROPOSAL

TEXT SUMMARIZATION & INFORMATION RETRIEVAL IN INSURANCE DOMAIN USING NLP

SHRADDHA MALADKAR

Research Proposal

FEBRUARY 2023

Abstract

Businesses are the backbone of an economy and rising competition is a fundamental driver of productivity and output growth. Competition can be considered healthy for business as it keeps the business on its toes and makes it imperative for it to innovate and improve. This creates a need for benchmarking the competition as benchmarking is a method of understanding company's performance against others and ensure that they can stay competitive in the market. There are different ways through which benchmarking can be performed like keeping an eye on competitor's social media, press releases, market trends and extract key information from all the data that is gathered by various sources. Traditionally, this is done via manual intervention that is reading and going through this huge amount of data, retrieving relevant information which is time consuming and prone to human error affecting the productivity of the business. Now these tedious tasks can be easily accomplished using NLP (Natural Language Processing) with techniques like Information Retrieval (IR) and Text Summarizer. Training models using these techniques makes productive summary generation probable and retrieve key information. There are two methods which are used commonly to generate summaries where extractive method scans the article to discover the related sentences while isolating only that piece of information from the article whereas abstractive technique understands the text and then generates the summary. Abstractive process is considered to be a bit more complex, hence transformer-based pretrained models are employed to compare the content & creating the summary. The study uses the data obtained by web scraping news articles and public disclosures of the competitors of Insurance domain to assess and compare the outcomes found using the machine learning models. The crucial information found out by the model can be then used to get insights into competitor's business and help the company determine the scope of improvement and devise strategies that work best for the business.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. Background

Before the internet was discovered, competitive intelligence was majorly conducted through industrial espionage. While the latter still occurs, there are much simpler and more ethical ways to gain insights on the competitors' business [10]. By being attentive to the data competitors and their customers leave on the internet, the company will be able to make well-informed strategic decisions.

Competitive intelligence also known as market intelligence is the action of gathering and analysing information about products, market, customers and competitors. Through this method, one can expect competitor's likely next steps, spot possible threats & opportunities, and gain competitive benefit. By collecting this valued information through competition benchmarking makes it easier for the company to make an effective and competent competitive strategy [10].

Benchmarking is a way of figuring out company's accomplishments against others and more importantly guarantees that they can stay ahead in the market. It offers an insight into what a competitor is achieving, what works, what doesn't work, and something that companies might use for their own progress, whether it's achieved by adopting effective practices or dodging ineffective ones.

The reason why competitive intelligence is so significant is because it helps businesses understand their position within the competitive landscape, recognize opportunities, identify their market share along with many challenges in the industry. And to do this, a brand needs to understand how they measure up against the market and collect some data on other brand's strategies and with no helpful way to contextualize the data, the company is stuck inspecting data in vain. This results in lagging of company's social strategy thus proving ineffective [12].

Traditionally, competitive benchmarking is very long and laborious process which is particularly time consuming for teams to implement and frequently found with human errors. One needs some data-driven insights or else it's very tough to quantify what accomplishment looks like compared to other companies in the market or to get a clear picture on top-notch trends. And if real-time data is missing, one can neither form effective strategies nor can they advise customer care services, social media, PR teams and marketing teams [12]. Now this competitor's data can be extracted from various sources like news article, social media posts,

press releases, official product launches and public disclosures using techniques like web scraping.

Web scraping is method that can automatically extract huge amount of data from any web pages. Majority of the extracted data is unorganised and in HTML format which can be transformed into structured data, stored into a database and can be made useful in many applications. There are numerous ways to do web scraping such as using software that are readily available online, particular API's or even generating a code for web scraping using any language like Python, Java etc. Some websites including Twitter, Facebook, Google etc. exposed their API which permits any user to gain access of the data in organized format. But there are some sites that blocks user's attempts to gain access to huge data available in organized format or they may lack technologically advancement. Such situations arise a need to scrape data from any websites [22].

Now once the data is gathered which is usually huge in size needs to be processed and then analyse to draw insights and use it to the company's benefit. This is where NLP techniques like Information Retrieval (IR) and Text Summarization comes into picture.

Information retrieval is the process of retrieving the most accurate information from text which is usually based on a specific query provided by the user using context-based indexing or metadata [23].

NLP text summarization is the method of breaking down long texts into consumable paragraphs or sentences. This method separates vital information while conserving the meaning of the text. This decreases the time required for grasping prolonged pieces such as articles without the loss of any crucial information [24].

2. Problem Statement

The amount and difficulty of unstructured data is increasing exponentially and this brings forward some new challenges. More the time spent on manual data extraction means less time spent on analysis which creates delays in drawing insights and informing those insights to stakeholders. Also, there is always a possibility of human error or poor data quality with doing things manually [12].

Currently gaining information on competitors is done via visiting their official websites, finding documents, press releases, news articles and social media presence featuring the competitors. One needs to pro-actively keep an eye on competitor's performances, product launches, news updates as well as keep track of market trends, manually collect data from multiple sources and for all the competitors that are present in the market. Then information is extracted from such documents/news articles/social media posts by reading each and every line and then it is summarized by an individual.

This manual intervention makes this process vulnerable to human error affecting the accuracy of the results. There are chances where some of the information might be overlooked which would have been important for business.

To overcome above mentioned challenges, I am proposing a system with -

- **Web Scrapper tool** which will be used to gather the competitor's data and market trends using Web Scraping which will help us gain real time data.
- **Text Summarization & Information Retrieval tool** which will extract relevant and important information and also condense the news articles/ social media posts into a shorter version and preserve important contents

Initially, scores for every sentence were calculated using statistics where sentences with higher scores were chosen. Various methods were utilized to compute this score like TF-IDF [4] and Bayesian models [5]. Drawback of this approach was that these were extractive methods and were truncating the text even though they managed to generate a thorough summary by key phrase extraction. Then Bayesian Learning Models [3], a machine-learning approach was used. These models were effectively aimed at pattern recognition in the given texts while finding a correlation amongst various words. Due to importance of the sequence of words in natural language understanding and development, each sentence can be considered as sequential data.

In this scenario where one needs to handle sequential data, the architecture needs to remember information using some memory [3].

RNN [6] and the LSTM network [8] preserves sequential info via connected nodes by storing important information while overlooking irrelevant information which was used in summary generation. In developing an encoder-decoder model, LSTM network was used.
Models based on Seq2seq [7] architecture which were executed using the encoder-decoder framework in order to resolve NLP problem statements presented promising outcomes, but still there was the problem of parallelization. Even after the sequential information is preserved the processing is done by taking one input at a time. This creates a tricky situation despite the model obtaining better outcome, it failed for every possible scenario [3].

In the text summarization systems, many challenges occur such as detecting the topic, analysing, creating a summary, and assessing the output summary. Many text summarization tools have been built on extractive technique. Hence, now there is a need to explore the area of abstractive text summarization which can create automatic summaries. It gives a way to tackle the enormous rising information found online. It also aids in finding the most relevant information among the news articles or any document for that matter, quicker along with producing a precise summary which can be considered as primary purpose of this research.

3. Aim & Objectives

The aim of this research is to build an information retrieval tool that extracts key information form the documents and a text summarization tool which focuses on the issue of identifying the most significant parts of the text and generating coherent summaries from the news articles.

The goal of this study is to create a social listening tool that monitors competitor's actions and provides only vital information which can in turn help devise effective strategies and achieve optimal growth of the business.

The research objectives are formulated based on the aim of this study, which are as follows:

- To collect data using Web Scraping, combine the data extracted from multiple sources and perform data cleaning operations
- To implement Information Extraction technique to extract information from the documents
- To create an optimized and efficient algorithm in order to produce a text summary of the news articles
- To evaluate the performance of the techniques/algorithms to achieve optimal results

4. Significance of Study

There is an exponential growth in the daily generated data as the cost of storing the data is reducing and computational power of the systems is increasing. But any means to extract the information and be able to query it is lacking then the information collected is useless. Information is knowledge only if we are able to extract relevant information. It takes time as well as efforts to read an entire article, analyse it, and bifurcate the important concepts from the raw text. At least 15 minutes are required to read a 500-word article while automatic summary tool summarises texts of 500-5000 words in mere seconds. This allows the user to read less data and still be receiving the key information and making sound judgments. Instead of reading entire news stories that most of the time contains unrelated information, summaries of such websites can be accurate with only about 30% of the original article's size. These tools in turn can help companies in numerous ways, elimination of manual intervention to extract data/information will provide more accurate data with reduced amount of time. This reliable data will help in making strategies and taking business decisions which can lead to greater profits. Keeping track of the competitor's action will benefit in finding drawbacks in their own products/services and improve the quality of the same.

5. Scope of the Study

The scope determines the level up to which the research can be discovered and how the methods/techniques can be bound to specific constraints.

1. Using web scraping technique, the news articles/ documents of the competitors will be gathered which will be used to develop Information retriever and text summarizer.
2. Finding out various methods/algorithms by understanding NLP techniques that has been developed over the years to build a text summarizer, identify all the drawbacks and then work on a summarization & IE tool which can overcome said drawbacks.
3. Evaluating the model using ROGUE and making sure that the model gives an accurate measure of performance that model classifies most of the words (recall) and while doing so with no irrelevant remarks (precision).

The research will not only benefit the insurance domain but can also be used to benchmark the competition in any domain. Apart from this, there are many other applications of the Text Summarizer where lengthy documents need to be summarized or where key information is to be captured.

6. Research Methodology

6.1 Dataset and its description

The dataset being used for this study is gathered from multiple sources and public disclosure data was downloaded from official websites of the competitors. Information will be retrieved from these documents that are relevant to the business. News articles were extracted using web scraping from article links which were extracted from SERP API news search using some keywords. As this study involves competition benchmarking of Insurance domain, publicly available data in the form of news articles, public disclosures, press releases, product launches of companies like HDFC Life, ICICI Prudential, LIC, Max Life etc were obtained along with news articles covering market trends, innovations in insurance industry etc

Table 1. Dataset with column description

Sr No.	Column Name	Column Description
1	Title	Headline of the News Article
2	Links	Link of the Article
3	Date	Date on which the article is published
4	Source	News channel name
5	Article_1	Content of the article scraped using web scraping method 1
6	Article_2	Content of the article scraped using web scraping method 2

6.2 Data Pre-processing

Data pre-processing is one of the important steps for any NLP problem statements. There are multiple files generated which needs to be combined and checked for any missing or null values. Tokenization is a major step in my approach. Every sentence and word must be converted into tokens and calculate the maximum and minimum length of all the Articles.

6.3 Model Understanding

Text summarization process includes following two ways-

- Extractive technique - In extractive summarization, from the text fed as the input, a summary is produced by choosing a subsection of the entire sentence. After this, the

most significant phrases or sentences are recognized and chosen based on a score that is calculated based on the words present in that sentence [3].

- Abstractive technique - This type of summarization provides the abstract summary of the article by extracting the most critical facts from the text. It generates one-of-a-kind summary. [2].

The conventional and cosine similarity approaches use extractive type technique. Creating abstract summarization model is a problematic piece of work as it suffers from problems such as semantic representations. So, this study will concentrate on abstractive summarization in order to obtain a precise and fluent summary as its more challenging. It also might provide better results as compared to other techniques. In order to perform abstract type technique, an encoder-decoder neural network with an attention model will be built. [2].

Some pre-trained language models like BERT [15], PEGASUS [16], UNiLM [17], GPT [18] has revolutionized the field of Natural Language Processing with their outcomes and have inspired many others to contribute for the summarization of text. Transformer architecture which was developed by Google [15] in 2017 has proved to be very significant for most of the pre-trained models. Encoder-decoder framework which is kind of similar to RNN, is the building block for Google's architecture. Transformer model [19] was designed to solve NLP tasks that consisted of converting an input sequence to an output sequence. Hugging face library [20] provides some of these pre-trained language models which aids in resolving many problem statements [3].

Transformers based pre-trained models - Hugging face [20] is an open-source which gives several NLP libraries and datasets including the Transformer library. This library comprises of numerous pre-trained models in order to generate summaries which can easily be tweaked for any given dataset.

Pipeline: The pipelines are used by various pre-trained models for inference. It presents a simple API dedicated to some tasks like text summarization by abstracting majority of the library's complex code. Pipelines contains techniques like Tokenization, Decoding and Inference which maps every token in a representation that makes sense. Considering English language, these transformers summarization pipeline from Hugging face has been proven to make the job simpler, effective and more competent to execute. [3]

BART: Bidirectional and Auto Regressive Transformers [21] constructed using a seq2seq model which was trained with denoising as a pre-training purpose. After merging a BERT-like encoder [15] and a GPT-like decoder [18], a typical seq2seq model architecture is used. In the pre-training task randomly rearranging of the phrases takes place and some new arrangement where text ranges are switched via a single mask token. Although its similar to the BERT [15] model and considering comparable sizes, BART has around 10% additional features than a BERT model. The decoder is autoregressive which is controlled to create sequential NLP projects like text summarizer. [3]

Basic Understanding of Transformer model

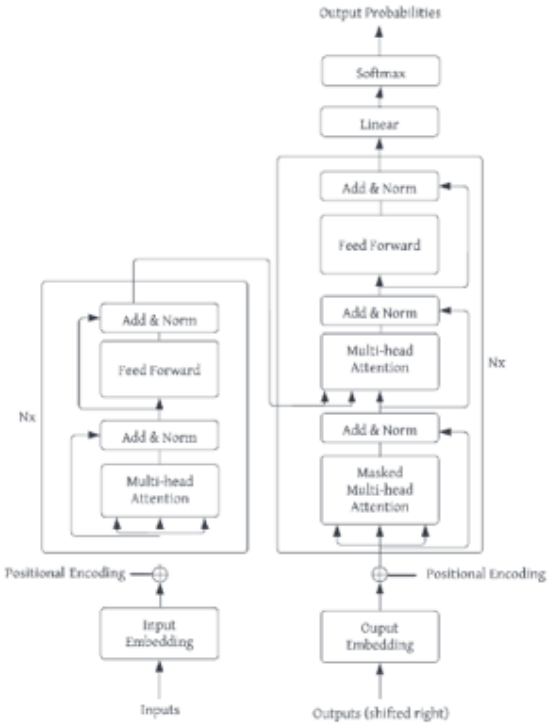


Fig. 1. Model Architecture of Transformer

A "text-to-text" format where the model is inputted with some text while output is generated in text format has proven to be the solution to majority of NLP problem statements. Some of the most successful neural sequence transduction models implemented this similar architecture.

The said model is automatically regressive on every step, along with the formerly produced characters which are added as input when progressing the subsequent step [2].

1. **Encoder:** It comes with a pile of N = 6 homogeneous layers where every layer has a multi-head mechanism which provides self-attention with a fully connected feedforward network based on position. A residual connection is used around the two sub-layers, shadowed by layer normalization. This gives the output of each sub-layer as LayerNorm (x + Sublayer(x)), where Sublayer(x) is the function executed by the sub-layer.
2. **Attention:** This function is defined as mapping an input query with a set of key-value pairs (which are vectors) to the output. The output found is a weighted sum of values, where the weight allocated to each value is recognized by a compatibility function of the query with the respective key. Main benefit of using multi-head attention is it lets the model share info from various representations subspaces at many positions. In case of a single attention head, averaging is used to avoid it [2].
3. **T5 Approach:** Text-to-Text Transfer Transformer, is an architecture based on transformer that utilizes a text-to-text approach. It is a pre-trained model consisting of an encoder and decoder which is trained using both unsupervised and supervised tasks. Multi-task learning concept is used where each task is converted into a text-to-text format. [2].
4. **Attention Masks:** Different architectures applied for processing the data vary with respect to the attention masks. Transformer uses a sequence as an input and creates a new sequence as an output where length of both the sequence is alike. This is processed during self-attention operation in the model. Altogether, values of the output sequence are created by calculating a weighted average of the input sequence [2].
5. **Decoder:** It contains a pile with N = 6 homogeneous layers. On top of two sub-layers, it joins a third sub-layer which offers multi-head attention to the output of the encoder. A constant connection around the two sub-layers is applied, succeeded via a layer normalization. In order to avoid the positions from concentrating on the subsequent positions, an updated self-attention sub-layer is applied [2].

6.4 Evaluation Metrics

Model evaluation is an important task in any development. Manual and semi-automatic assessment of huge models used for text summarization is very costly and troublesome. There has been a lot of work done in order to employ automatic metrics which enables a faster and economical evaluation of models.

Recall-Oriented Understudy for Gisting Evaluation also known as ROUGE is an automatic summary benchmarking metric. The package provides a set of automated metrics based on the lexical bunch among the summary generated by model and reference summary [2]. ROUGE scores are computed by finding out the overlapping words quantity between the model-generated summary and the reference. I will be using ROUGE as my model assessment method.

Types of ROUGE are-

1. ROUGE -N: This metric specifies the overlapping of *n-grams* among the summary created via model and the reference summary. Below equation denotes this-

$$ROUGE - n = \frac{\sum_{S \epsilon RS} \sum_{gram_n \epsilon S} Count_{match}(gram_n)}{\sum_{S \epsilon RS} \sum_{gram_n \epsilon S} Count(gram_n)} \quad \ldots\ldots (1)$$

- reference summary is a set denoted by RS
- length of $n-gram$ is denoted by n
- maximum number of $n-grams$ co-occurring in the output summary for a set of references is denoted by *Count match (gram n)*

2. ROUGE -L: L denotes the Longest Common Subsequence (LCS) matching between the reference and the output summary.

3. Recall: This parameter calculates the number of coincides $n-gram$ occurring in the reference and the summary created by the model. The number generated is then divided by the total number of $n-gram$ present in the reference.

$$\frac{number\ of\ n-grams\ found\ in\ model\ and\ reference}{number\ of\ n-grams\ in\ reference} \quad \ldots\ldots (2)$$

4. Precision: This parameter is computed by taking model's $n-gram$ count into consideration rather than the reference $n-gram$ count as done in the recall.

$$\frac{number\ of\ n-grams\ found\ in\ model\ and\ reference}{number\ of\ n-grams\ in\ model} \quad \ldots\ldots (3)$$

5. F1-Score: Recall and precision values are used to calculate ROUGE F1-score.

$$2 \times \frac{precision \times recall}{precision + recall} \quad \ldots\ldots (4)$$

A precise model performance can be achieved which depend on the model that recognizes abundant words (recall) and can be done without creating irrelevant remarks (precision) [2].

7. Required Resources

7.1 Hardware Requirements

For this research, Hardware with following specifications will be used-

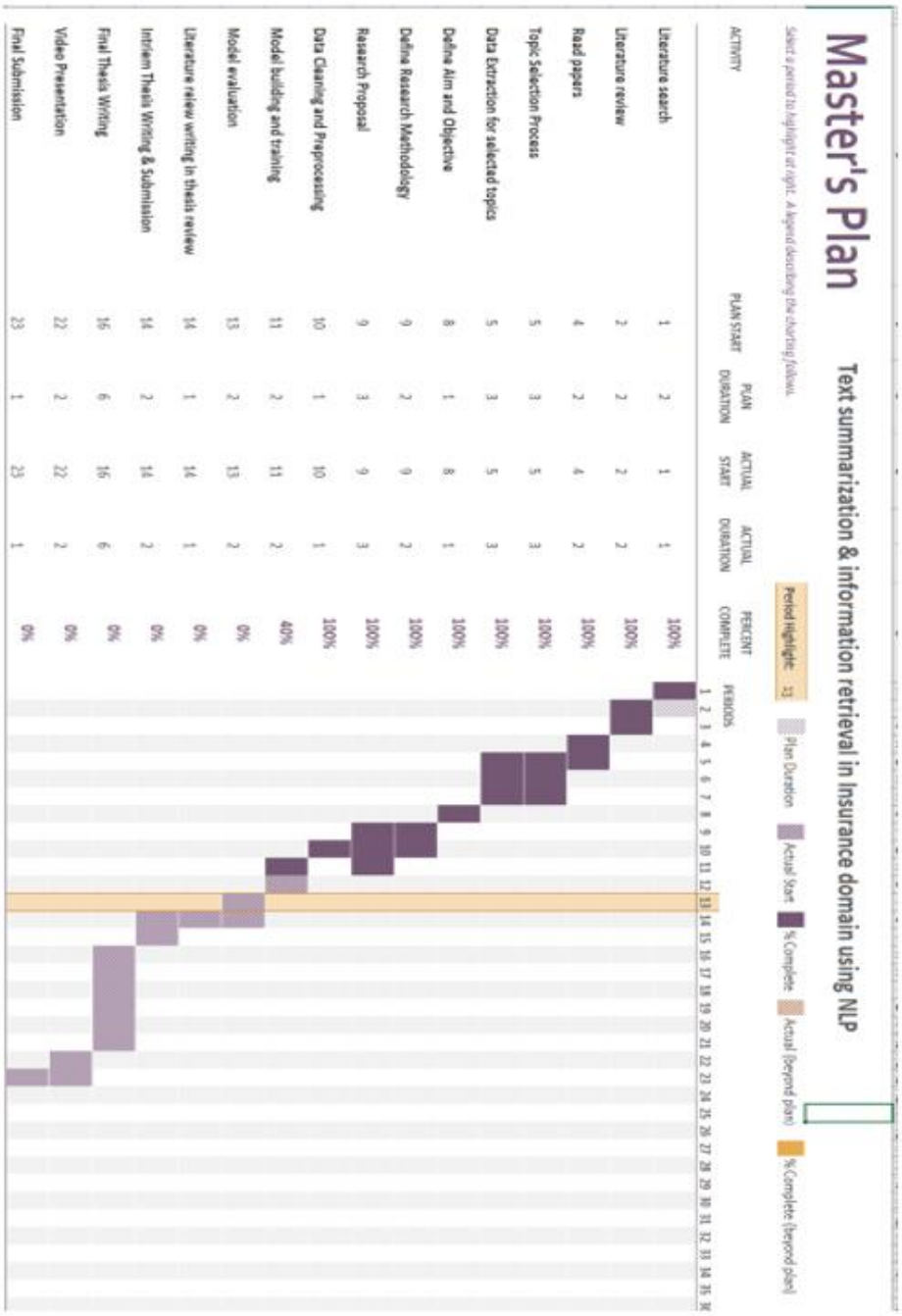
- Device: Laptop
- Processor: AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz
- System type: 64-bit operating system, x64-based processor
- Wireless Networking: 802.11n
- RAM: 24.0 GB
- GPU: Nvidia GeForce RTX 3060 6GB

7.2 Software Requirements

Software/Libraries/OS specifications that will be required are-

- OS: Windows 10
- Python version: 3.10
- Code Editor: Visual Studio Code/Anaconda/Jupyter NB
- Libraries: Pandas, Numpy, Sklearn, NLTK, TensorFlow, Keras, transformers etc
- Other Software: Microsoft Office (Excel, Word, Power Point, One Note), Google Chrome, Adobe Acrobat Reader, Notepad, 7-Zip, Tableau etc

8. Research Plan



References

[1] Sethi, P., Sonawane, S., Khanwalker, S. and Keskar, R.B., 2017, December. Automatic text summarization of news articles. In 2017 International Conference on Big Data, IoT and Data Science (BID) (pp. 23-29). IEEE.

[2] Balaji, N., Megha, N., Kumari, D., Kumar, S. and Bhavatarini, N., 2022, October. Text Summarization using NLP Technique. In 2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER) (pp. 30-35). IEEE.

[3] Gupta, A., Chugh, D. and Katarya, R., 2022. Automated news summarization using transformers. In Sustainable Advanced Computing: Select Proceedings of ICSAC 2021 (pp. 249-259). Singapore: Springer Singapore.

[4] Christian, H., Agus, M.P. and Suhartono, D., 2016. Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). ComTech: Computer, Mathematics and Engineering Applications, 7(4), pp.285-294.

[5] Nomoto, T., 2005, October. Bayesian learning in text summarization. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (pp. 249-256).

[6] Graves, A., 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

[7] Nallapati R, Xiang B, Zhou B (2016) Sequence-to-Sequence RNNs for Text Summarization. CoRR abs/1602.06023:

[8] Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[9] Keneshloo, Y., Shi, T., Ramakrishnan, N. and Reddy, C.K., 2019. Deep reinforcement learning for sequence-to-sequence models. IEEE transactions on neural networks and learning systems, 31(7), pp.2469-2489.

[10] Kuhanec, T. (2022) What Is Competitive Intelligence?. Available at: https://www.meltwater.com/en/blog/competitive-intelligence.

[11] Jagtap, R. (2021) Abstractive Text Summarization Using Transformers - The Startup. Available at: https://medium.com/swlh/abstractive-text-summarization-using-transformers-3e774cc42453.

[12] What is Competitive Benchmarking? Definition & Guide (no date). Available at: https://www.sprinklr.com/blog/what-is-competitive-benchmarking/.

[13] Sharma, N. (2021) NLP Based Information Retrieval System - Towards Data Science. Available at: https://towardsdatascience.com/nlp-based-information-retrieval-system-answer-key-questions-from-the-scientific-literature-b8e5c3aa5a3e.

[14] Accern, T. (2022) NLP Text Summarization: Benefits & Use Cases. Available at: https://accern.com/blog/nlp-text-summarization/.

[15] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[16] Zhang, J., Zhao, Y., Saleh, M. and Liu, P., 2020, November. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In International Conference on Machine Learning (pp. 11328-11339). PMLR.

[17] Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.W., 2019. Unified language model pre-training for natural language understanding and generation. Advances in neural information processing systems, 32.

[18] Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I., 2018. Improving language understanding by generative pre-training.

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[20] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

[21] Wang, L., Zhao, W., Jia, R., Li, S. and Liu, J., 2019. Denoising based sequence-to-sequence pre-training for text generation. arXiv preprint arXiv:1908.08206.

[22] Web Scraping (no date). Available at: https://www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it/

[23] Information Retrieval (no date). Available at: https://www.analyticsvidhya.com/blog/2021/06/part-20-step-by-step-guide-to-master-nlp-information-retrieval/.

[24] What is NLP Text Summarization: Benefits & Use Cases (no date). Available at: https://accern.com/blog/nlp-text-summarization/#:~:text=NLP%20text%20summarization%20is%20the,articles%20without%20losing%20vital%20information.