Patent Semantic Annotation for Practitioners

By

Summiya Kabir

00000318255

MS CS-2K19

Supervisor

Dr. Muhammad Khuram Shahzad

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science (MS CS)

In

School of Electrical Engineering & Computer Science (SEECS) ,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(July 2023)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Patent Semantic Annotation for Practitioners " written by SUMMIYA KABIR, (Registration No 00000318255), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: MKhurram.

Name of Advisor: Dr. Muhammad Khuram Shahzad

Date: 08-Jun-2023

HoD/Associate Dean:_______________________

Date: ___________________________________

Signature (Dean/Principal): ________________

Date: ___________________________________

Approval

It is certified that the contents and form of the thesis entitled "Patent Semantic Annotation for Practitioners " submitted by SUMMIYA KABIR have been found satisfactory for the requirement of the degree

Advisor : Dr. Muhammad Khuram Shahzad

Signature: MKhurram.

Date: 08-Jun-2023

Committee Member 1:Dr. Muhammad Muneeb Ullah

Signature:

11-Jun-2023

Committee Member 2:Pakeeza Akram

Signature:

Date: 08-Jun-2023

Signature:

Date:

Dedication

This thesis is dedicated to my family, especially my beloved and brave mother.

Certificate of Originality

I hereby declare that this submission titled "Patent Semantic Annotation for Practitioners " is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: SUMMIYA KABIR

Student Signature: Summiya ______________

Acknowledgments

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. Who only has the power to honour whom He pleases, and to abase whom He pleases. Verily no one can do anything without His will. From the day, I came to the University of Sciences and Technology (NUST) till the day of my departure, He was the only one Who blessed me and opened ways for me, and showed me the path of success. There is nothing which can pay back for His bounties throughout my research period to complete it successfully.

Summiya Kabir

Contents

List of Figures

List of Tables

Abstract

Having a patent document, associating discrete semantic annotations has emerged as an exciting research area. Text annotation aids patent practitioners, such as; examiners and attorneys in quickly identifying the important arguments of any invention, allowing for quick marking of the patent text. In the manual patent analysis process, recognizing the semantic information by marking paragraphs is commonly used to improve readability. To overcome this time-consuming and laborious task, we have used Deep Learning models on the dataset of size 150k patent samples. Prior researchers have used different Machine Learning models but we approached this problem as deep learning task and investigate the merits of Deep learning models for highlighting patent paragraphs. To raise the level of performance of the base models of Patent Sentiment Analysis, we trained five different deep learning-based models on the Patent dataset: Simple Feedforward Neural Network, Long Short-Term Memory (LSTM), BERT, RoBERTa, and DistilBERT. We have trained different Feed Forward Neural Networks (FFNN) with different hyperparameters and logged our precision, recall and f1-score. We used pretrained GloVe embeddings as wordembeddings. Sentence embeddings were calculated as the element-wise mean of all word vectors of that sentence. We have trained FFNN for 500 epochs and log the loss and test the accuracy of the model after each epoch. The loss and accuracy graphs are smoothed using exponential moving averages so we can see the overall trend of the model. This work uses Deep Learning to assist patent practitioners in automatically highlighting semantic information and creating a sustainable and efficient patent analysis.

Chapter 1

Introduction

This chapter provides the introduction of the thesis study. Specifically, it includes a motivation for the subject, a declaration of the problem, and a discussion of the proposed solution, together with the thesis' primary contributions.

1.1 Motivation

The drafting of patent applications differs globally by area and also according to literature styles. Such as innovations or applications created in the Asian region frequently include specific headers or sections in a patent that detail the important elements of the invention. In other instances, they focus on describing the selling points of the invention with a separate heading like Advantages effects of the invention (AEI) [31]. These technical matters are frequently effective technological advantages of the invention. The same is true for other annotations that can be used to distinguish technical subject matter, such as Technical problems(TP) connected to prior efforts or any other boilerplate (simple descriptive information, problem-solving procedures (PSP)) [31]. Such portions or special sections make it easier to navigate through the patent documents and improve readability for the reader. However, not all patent documents contain such pre-defined annotations. The rush in the submission of patent applications in recent years has significantly increased the burden at evaluation centers to prosecute innovations.

In order to perform such prosecution efficiently, the evaluator must carry out a variety of tasks, including a search for the prior state of the art, an assessment of innovation within the parameters of patent law, and the provision of a critical review of the judgment in a form report document or hearing. Few approaches have recently shown an interest in aligning patent analysis tactics utilizing the range of DL methodology to handle complicated patent processes [35]. Simple automation for highlighting significant passages for assessment in such a complex procedure might make documentation and determining the inventions' non-obviousness easier. Several machine learning surveys in the patent industry have previously been published in an effort to promote multidisciplinary research. According to Joho et al, a descriptive study of the patent literature is provided, and they describe numerous criteria for patent searches and functionalities that were adopted during the study [5].

Text-mining and visualization methods for patent analysis are outlined by Abbas et al. [7]. Zhang conducted a survey of data mining methods for patent analysis and pointed out the technical difficulties involved in their work [11]. Using benchmark datasets, a different study that employed patent retrieval quantitatively compares various methods. [22]. The retrieval techniques that are addressed fall into the categories such as keyword-specific, semantic-based, pseudo-relevance feedback, metadata-based, and interactive techniques. However, deep learning methods to take on this issue are not offered, despite the retrieval tasks and datasets being discussed in detail.

Table 1.1 demonstrates that the major patent offices have taken steps to promote deep learning's application after realizing the potential of the technology.

As far as we are aware, there hasn't been any research on deep learning techniques for various patent analysis tasks. By presenting an overview of deep learning network designs utilized for diverse patent analysis tasks in this paper, we want to bridge this knowledge vacuum.

1.2 Problem Statement

Patents are monopolies given by the government for those innovations that are unique, innovative, and non-obvious. When organizations or individual inventors want to get a patent for an innovation, they must describe it in text form and as demonstrations in a patent application [27]. Traditionally, a patent attorney with experience in both law and technology will create such a patent application. It includes claims that outline the intended scope of the invention's protection. A patent will only be issued if the subject

Task	**EPO**	**WIPO**	**USPTO**
Patent classification	Uses CPC scheme to classify patents	Uses the IPC main class, subclass, or main group as the basis for automatic categorization	Uses chat bots for asking conceptual questions to help in automatic claim analysis and classification
Prior art search	Utilizes Gold Standard generation, automated annotation, searching, and query generation	Using AI-assisted tools, does cognitive or semantic search	Examine the entire document to the corpus of grants and pre-grants
Data interpretation	Build open source libraries to analyze data and technology trend analysis	strategic and economic evaluation	Enhanced patent analytics on a web browser
Patent analysis	Automatic annotation and exclusion detection	advanced big data analytics to enhance evaluation procedure	Utilizes advanced data analytics for statistics and text analytics to ensure patent quality
Image analysis	Automatic search for images and figures	Searching for patents using images from a global brand database	Searching images for patents and trademarks

Table 1.1: Patent analysis procedure

matter described in these claims is novel and innovative in comparison to the previous art.

Therefore, it is advantageous to do a quick prior art search even at this early stage of drafting a patent application to evaluate the likelihood of granting and ultimately to modify the claim wording. However, retrieving pertinent prior art i.e. old publications that are relevant to the innovation under consideration can be difficult. It's possible that earlier innovators who had a similar concept used different terminology to describe it. Consequently, a straightforward keyword search is not always beneficial. For a patent to be issued, the specified invention must be unknown or easily deduced from the so-called previous art, which includes any written or spoken publication available prior to the submission's filing date [18].

Any person seeking a grant for an invention must first create a patent application documentation at the patent office where a patent document undergoes a large examination process by the patent officers. This is when an examiner's role comes into play. Often times patent examiners need to carefully read the document to find the prior art. This can be done by searching the prior art in any patent databases, other patent applications, or any other patent sources. Because of database indexing, a rapid listing of documents is often possible. This is still a hard and time-taking process for examiners to manually mark important technical subject topics in a timely manner and make a decision on the application's inventive step. When patent applications undergo comparison with published works, examiners and attorneys often highlight text passages that could be important components for key points.

The highlighted data facilitates the examiner not only to write a thorough search report but also to win over any objections made by patent applicants in official hearings. Patent applicants or their agents, such as attorneys, on the other hand, go through a variety of patent documents to assist their clients. In such scenarios, it is necessary to keep track of the several benefits of the invention with respect to the prior art. Because this defines the scope of the patent, as the scope of the invention expands, the examination process gets more difficult and critical.

As a result of the examiners' reports, it may require numerous iterations of modifications. "Patent examiners use manual highlighting of text in documents. This provides us with an indication that highlighting patent paragraphs is an important sub-task in patent examination.

Therefore, we formulate the problem statement as follows, "manually highlighting paragraphs for legal artifacts in a complex legal document for attorneys and practitioners is a challenging task."

1.3 Solution Statement

The rush of patent applications in recent years has significantly increased the workload at examination offices to investigate innovations. In order to conduct such prosecution efficiently, the examiner must carry out a variety of tasks, including a search for the prior state of the art, an assessment of innovation within the parameters of patent law, and the requirement of a crucial assessment about a decision in terms of a report or hearing. Few techniques have recently shown interest in integrating patent analysis strategies while utilizing the range of DL technologies to handle complex patent processes.

In such a complex procedure, simple automation for highlighting significant passages for assessment might make documentation and determining the inventions' non-obviousness easier. To the best of our knowledge, there is no documentation in the literature that employed dataset and training models for patent paragraph highlighting based on deep learning. We have proposed a solution with deep learning techniques to achieve better accuracy and results.

We formulate the solution statement as follows, "Automatically scanning through a large number of complex legal documents by employing deep learning techniques for legal artifacts."

1.4 Contributions

The main contributions presented in the thesis are:

- We have requisite data from the United States Patent and Trademark Office (USPTO). The data was cleaned by lower casing, stop words removal and averaging of words. the removal of single word and null sequences is also done on the dataset.
- The state-of-the-art models employed are Feedforward Neural Network(FFNN), Long Short-Term Memory(LSTM), BERT, RoBerta and DistilBERT.
- To the best of our knowledge this type of work has not been addressed so we have applied Deep learning methods in our research prior patent paragraph highlighting work was done by machine learning.

1.5 Thesis Outline

The rest of the thesis is structured as follows; Section 1 explains the introduction, section 2 presents the detailed reviews of the past work, section 3 discloses the methodology we used especially the dataset and models, section 4 describes the performances of the state-of-the-art models' comparison in highlighting the patent sentiment analysis process, and finally, section 5 presents a discussion related to conclusions, limitations and future work.

1.5.1 Literature review

This chapter discusses the challenges involved in analyzing/examining patent analysis and the state-of-the-art methods that have been used for granting patents. It also presents the deep learning techniques that have been used for patent analysis and highlights patent paragraphs and the problem identities in these approaches.

1.5.2 Research Methodology

We offer the suggested methodology to address the problems mentioned in Chapter 1 of the thesis. We specifically give an overview of the key building parts of the suggested framework, which are the gathering and processing of data, model construction, model training, and model evaluation for the data.

1.5.3 Implementation and Results

The experimental design, its execution, and the related outcomes in our suggested framework are presented in this chapter along with a comparison of the outcomes corresponding to various data sets.

1.5.4 Conclusion and Future Work

We offer the thesis's conclusion and indicate a few potential future directions for the work done in this chapter.

Chapter 2

Literature Review

This chapter explains the value of analyzing the patent document and goes over the difficulties involved in granting a patent. We also examine conventional, machine learning-based, and deep learning-based methods for examining patent applications and granting patents.

2.1 Patent Analysis

Patent analysis is the process of examining and evaluating patents and patent applications in order to gain insights and information about a particular technology or field of innovation. This can be done manually, through the use of specialized software, or through a combination of both. Automation is the use of technology to automate a process or task, often with the goal of making it more efficient or faster. In the context of patent analysis, automation refers to the use of software or other tools to automate parts of the patent analysis process, such as searching for relevant patents, analyzing their content, or organizing the results. This can help to streamline the process and make it more efficient and effective.

2.2 Patent Lifecycle

The patent life cycle refers to the process a patent goes through from the time it is filed until it expires. This process typically includes the following stages [42]:

- The patent is filed with the appropriate intellectual property office.

- The patent application is reviewed by a patent examiner to determine whether it meets the necessary criteria for the grant of a patent.
- If the patent is granted, it is published.
- The patent owner may enforce their rights by suing anyone who infringes on the patented invention.
- The patent may be renewed periodically to maintain its enforceability.
- Eventually, the patent will expire and the invention will enter the public domain, where it can be freely used by anyone.

The length of the patent life cycle can vary depending on the type of patent and the jurisdiction in which it is filed. For instance, in the United States, a utility patent's term (which protects the majority of inventions) is ordinarily 20 years from the date the patent application was submitted. The period for design patents is shorter, at 14 years after the date of grant [37].

2.3 Challenges in Manual Patent Analysis

There are several challenges that can arise when conducting a patent analysis. Some of the most common challenges include:

- Understanding the technical language used in patents is the first step toward evaluation. Patents often use technical and specialized language that can be difficult for non-experts to understand. This can make it challenging to accurately interpret the information manually contained in a patent.
- Identifying relevant patents is a hard task in a manual examination of patents as there are millions of patents filed every year, and it can be difficult to identify which ones are relevant to your analysis. This can be especially challenging if a person is not familiar with the specific technology or field of research under study.
- Determining the scope of a patent is the most important yet difficult part as patents are complex legal documents that may be challenging to interpret. It can be challenging to determine the scope of a patent, including what it covers and what it does not cover.

- Assessing the strength of a patent is important when conducting a patent analysis. This can be difficult to do, as there are many factors that can affect the strength of a patent, including the quality of the claims and the prior art in the field.

- Keeping up with changes in the law is difficult as the field of patent law is constantly evolving, and It can be difficult to stay updated with the most recent changes. This can make it difficult for the examiner to accurately assess the value of a patent or to understand the implications of a particular patent.

Taking into account all of these problems, manual patent analysis can be a time-consuming and labor-intensive process. It can be difficult for a single person to thoroughly review a large number of patents, and the complexity of some patents can make it challenging to accurately understand their contents and implications. Additionally, manual analysis can be prone to human error, which can lead to inaccurate conclusions. Some other challenges of manual patent analysis include the need for specialized knowledge and expertise, the potential for bias or subjectivity in the analysis, and the difficulty of keeping up with the rapid pace of technological innovation and the associated influx of new patents.

2.4 Traditional Approaches for Analyzing Patents

Finding valuable prior work is a difficult and time-consuming process, even for highly skilled professionals. Patent officers rely on current information systems to assist them with their work because of the vast amount of literature that must be taken into account and the necessary subject knowledge. However, the results of a prior art search, whether to determine a patent's patentability or validity, remain flawed and prejudiced based on the patent examiners and their search methodology [40]. However, various patent offices may get distinct conclusions from a single search. We expect that this research will pave the way for a qualitative and systematic analysis of search methodology.

Traditional approaches for analyzing patent documents typically involve manually reading the documents and extracting relevant information, such as the background, the problem the invention is trying to solve, and the key features of the invention. This information can then be used to understand the technology and assess its potential value. Another common approach is to use keyword search to identify patents that are relevant to a particular topic or technology. This can help to quickly identify trends and key players in a particular field.

In literature, information retrieval techniques like rule-based mining or keyword search are used to undertake patent annotation [11], [10], [1], supervised learning [4]. However, it is noted that there has not been any recent interest in utilizing machine learning to highlight patent paragraphs. In their proposal, [8] suggested using annotation to extract effect classes from patent abstracts and extending the dataset only with a small number of labeled examples. The state-of-the-art has associated limitations, such as manual moderation to keyword extraction and partially automated techniques to improve the creation of supervised data for training with labels.

The literature described above, in contrast to our approach, solely relies on information extraction from text based on its syntax and semantics. However, we suggest using the context of the text to determine the crucial subject topics in patent material, both at the sentence and paragraph levels. Additionally, other established markup-based and rule-based techniques for annotating patent papers have been presented [3]. This was somewhat helpful in identifying patent metadata. However, these rule-based and markup-based techniques have trouble locating the arguable issues and contextual aspects of patents.

Based on patent literature Semantic annotations based on ontologies are another type of annotation [25]. Once more, such an Ontology-based strategy is dependent on a manually constructed starting set of patterns. The outcomes of ontology-based approaches in recognizing contextual features are not much optimal. Some of the premium solutions, including PATSEER3 and Patsnap4, offered to automatically highlight patent content. However, this is only achievable if the solutions are not open-sourced and users are aware of the initial list of keywords.

As their name suggests, they solely emphasize keywords, although the vocabulary in patent literature is quite rich and diversified. Oftentimes, new terminologies are frequently created by patent applicants, and keyword-based techniques frequently focused less on atomizing the activities associated with patent analysis. Private IP specialists have recently shown interest in using Artificial Intelligence to highlight patent texts using IPGoggles5. The Artificial Intelligence algorithms being offered are trained on broad English literature, but there is no proof that the tools would be made public. To the best of our knowledge, the annotation process of highlighting patent paragraphs lacks trained algorithms and open-sourced datasets.

2.5 Machine Learning-based Approaches

A form of artificial intelligence called machine learning parses data using algorithms, learn from it, and makes informed decisions based on that data. When it comes to analyzing patents, machine learning algorithms might be used to analyze large amounts of patent data and identify patterns or trends that would be difficult for a human to spot[2]. For example, a machine learning algorithm could be used to identify common features or characteristics of successful patents or to predict which patents are likely to be approved by a patent office.

Additionally, patents can be categorised using machine learning algorithms in a variety of ways, or to group patents together based on their similarity. This can help patent analysts quickly and efficiently sift through large amounts of data to identify key trends and insights. The next section explores various machine learning-based methods for analyzing patents:

2.5.1 Natural Language Processing (NLP)

This approach uses machine learning algorithms to analyze the text of a patent and identify important concepts, keywords, and relationships between different ideas. This can be used to classify patents into different categories, identify trends and patterns in patent data, and even predict future developments in a particular field[2].

Various machine learning-based techniques have been employed to analyze patent documents. Researchers have recently begun using text-based methods (based on things like the patent title, abstract, keywords, or claims) to try and more nuancedly define the technologies covered by a patent in order to quantify peer-to-peer similarity and map technology landscapes and evolution[25]. Numerous approaches have been devised to tackle this matter, encompassing (i) keyword-based methods, (ii) ontology-based analysis, (iii) investigation of the SAO structure, and (iv) machine learning-based techniques. Keyword frequency and co-occurrence metrics form the basis of keyword-based approaches. Owing to their simplicity and lucidity, this strategy has been widely employed in the past.

Nevertheless, a significant drawback of keyword-based techniques is that they do not take into account the connections between concepts that are similar but are expressed using distinct terms. Certain studies utilize the subject, action, and object (SAO) structure found in patent documents as their semantic representation. To enhance the grammatical and meaningful structure, they adopt the SAO- methodology [13]. In the past, the majority of approaches relied on keyword-based methods. However, more recently, the distinctive characteristics of patent text, like the frequent application of synonyms and domain-specific technical jargon, led to a gradual transition towards embedding-based NLP methods.

2.5.2 Network Analysis

This approach involves using machine learning algorithms to analyze the network of relationships between different patents, inventors, and organizations. By examining the connections between patents, it is possible to identify important clusters of related ideas and identify key players in a particular field. NA has a number of potential uses particularly in the context of patent data. Every patent system in the world demands a search to find the previous art in the particular field of the invention in order to assess patentability criteria like obviousness and inventiveness [6]. The references (citations) in the published patent and the patent application are the publicly available documents taken into account in this search.

These references may be made to already-issued patents, pending patent applications, or non-patent literature like scholarly papers. With this, it is possible to build an analyzable network based on the references found in patents [30]. However, there are several restrictions on using patent data. It is not easy to create and analyze a patent citation network, and decisions and presumptions might affect the observations and results, so they should be carefully considered.

2.5.3 Sentiment Analysis

This approach uses machine learning algorithms to analyze the tone and sentiment of a patent's text, in order to identify positive or negative sentiment towards particular ideas or technologies. This can be useful for understanding the market potential of a particular patent, or for identifying potential obstacles to its commercial success. Patent paragraph highlighting falls under the category of patent sentiment analysis and information retrieval and is done previously using a variety of ML techniques [31]. Earlier studies have proposed a new dataset to train machine learning (ML) algorithms with the purpose of automatically identifying patent paragraphs according to various subject matter types. Subsequent researchers have the opportunity to build upon this work and enhance it further by incorporating initial baseline ML models that utilize the dataset[2].

2.5.4 Predictive Modeling

With this strategy, machine learning algorithms are used to create prediction models that, based on historical patent data, can predict future advancements in a specific industry. These models can be used to identify promising areas for investment or to identify potential competitors and collaborators. When a model is trained to create predictions using data, it is referred to as predictive learning, a sort of machine learning. In the context of patent analysis, predictive learning could be used to predict the likelihood that a particular invention will be granted a patent or to predict the potential market success of a patented technology[14].

The model would be trained on data about past patents and their outcomes and could be used to make predictions about new patent applications. Big data technologies are gaining a lot of interest because of their capacity to analyze vast quantities of diverse data sources and draw out valuable information from them. Big data technologies are now used in a variety of industries, including retail, marketing, and social media, as instruments for prediction in addition to being a methodology for studying the existing situation [20].

Overall, machine learning-based approaches for patent analysis can provide valuable insights into the state of a particular field and can help organizations make better-informed decisions about their patent portfolios and investment strategies.

2.6 Deep Learning-based Approaches

Deep learning is derived from machine learning that leverages neural networks to learn from extensive datasets. In the realm of patent analysis, deep learning approaches prove valuable for automatically classifying and analyzing patents based on their content. This can entail utilizing natural language processing methods to extract crucial information from patent filings such as the claims, abstract, and description, and using this information to train a deep learning model to classify patents into different categories or to identify key features or trends. Deep learning can also be used to generate summaries of patent documents or to identify relationships between different patents. Overall, the use of deep learning in patent analysis can help to automate and improve the efficiency of the patent analysis process.

2.6.1 Deep Learning Basics

Deep learning can be thought of as learning how to transform input data into output data. [6] This is a notable shift from conventional machine learning. In addition, the supervised machine learning method and deep learning both use the same fundamental building blocks [24].

Input Data The input data of a patent document can be text, photographs or references from patents filed and outside sources such as court records or citations.

Expected output data is task-specific and, with one or a few variables, could be discrete or persistent. For example, a classification procedure may lead to many class labels such as the classifications which should be included in patents.

Model Upon completing the learning process, a model is generated, which can take the form of a complex neural network or a naive Bayes classifier, among other possibilities. This model is the result of the training and can be used for various tasks, including classification and prediction. Based on input and expected output values, the model parameters are adjusted.

Metric Determined by task-specific loss functions, it is necessary to have a way to evaluate the model's progress, such as the proportion of accurate classifications. Deep learning, a subset of machine learning, employs multi-layer artificial neural networks. Learning is the process of altering a network's parameters, such as the layer weights, in a way that reduces a loss function for a collection of training samples or the training dataset. To achieve a local minimum of the function using stochastic gradient descent, the loss function must be differentiable. In contrast to unsupervised learning, supervised learning uses pairs of inputs and expected outputs that have been designated as ground truth as the training data. The word "deep" in deep learning refers to the number of stacked layers that are employed, primarily in computer vision applications, to create a hierarchical representation. Neural networks operate on the fundamental principle of learning representations from input data through layered transformations, facilitating a mapping from input data to output data[35]. Methods of deep learning developed from simpler neural networks. Convolutional neural networks are the most recent advancement in artificial neural networks, which have a history dating back more than 70 years. Neural networks function in a similar manner to other traditional machine learning algorithms and can be applied to comparable tasks. Like support vector machines or decision trees, they necessitate numerical (hand-crafted) features as input. Consequently, the foundational artificial neural networks share similarities with these techniques. However, these fundamental neural networks laid the foundation for contemporary deep learning, and various research publications use neural networks to complete tasks including patent analysis.

There are several approaches to using deep learning for patent analysis. Some common approaches include using convolutional neural networks (CNNs) for analyzing and extracting information from patent documents, recurrent neural networks (RNNs) for natural language processing of patent text, and graph neural networks (GNNs) for analyzing the relationships between different patents and inventors. Other approaches may include using transfer learning to fine-tune pre-trained models on patent data, or using generative adversarial networks (GANs) to generate new patent ideas. Ultimately, the specific approach will depend on the specific task at hand and the goals of the patent analysis [39].

2.6.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Network (CNN) is a kind of artificial neural network that is extensively used in image recognition and computer vision. In the context of patent classification, a CNN may be used to analyze and classify images contained in patent documents, such as drawings or diagrams. It can also be used to extract features from text in patent documents, such as the descriptions of inventions. The use of a CNN can help automate the process of patent classification and improve the accuracy of the results [45].

In recent times, deep learning approaches, notably convolutional neural networks (CNN), have made significant advancements in sound recognition, image processing, and speech recognition; nevertheless, patent categorization has not yet benefited from these advancements. A deep learning technique for classifying patents based on CNN and word vector embedding was proposed in a study named DeepPatent, where they compared it to other algorithms in the CLEF-IP competition and evaluated the algorithm on the common patent classification benchmark dataset CLEF-IP. In tests, DeepPatent with automatic feature extraction surpassed all other algorithms using identical data for training, achieving a classification precision of 83.98% [15].

The probability analysis method forms the foundation of conventional patent analysis. An innovative method to extract features from the patent text was put forth in another research and was based on a text-processing Convolutional Neural Network (CNN) that had been improved. It combines text mining and vector space modeling (VSM) methods in order to identify the abstract topics that exist in a collection of patent filings. In order to build a sort of organized data of patent text for use in subsequent patent analysis, it then maps the original data to a dataset of a high-dimensional vector space. [16].

2.6.3 Recurrent neural networks (RNNs)

Recurrent neural networks (RNNs) are a specialized type of artificial neural network renowned for their proficiency in handling sequential input, such as time series data or natural language. They are referred to as "recurrent" because they use feedback links in the network to convey information from one phase in the sequence to the next. In the context of patent analysis, RNNs can be used to examine and classify patent documents based on the content of the text [12]. This can be useful for tasks such as identifying relevant patents for a particular technology or determining the novelty of a new patent application.

To use an RNN for patent analysis, the patent documents would first need to be preprocessed and transformed into a suitable input format for the network. This might involve tokenizing the text and converting it into a sequence of word embedding, for example. The RNN would then be trained on a labeled dataset of patent documents, using supervised learning techniques to learn to classify the documents based on their content. Once trained, the RNN can then be used to classify new patent documents based on their content. This can be useful for tasks such as identifying relevant patents for a particular technology or determining the novelty of a new patent application.

Recent research has focused on applying deep learning to enhance the performance of patent quality classification. A significant problem in the realm of patent analysis is the detection of patent quality, which can offer valuable information for business and industrial decision-making. A patent's quality can be assessed using a variety of criteria, including the prediction system for invention protection, market value, technical skill, leadership position, and others. In one study, a deep recurrent neural network (DRNN) model was developed as a powerful feature extraction method to recover concealed features from the unprocessed text of a patent document. A gradient-based neural network with four functions, including cross-entropy loss computation, word embeddings, DRNN, and a fully connected neural network, is used to build the proposed patent quality rating system. Comparing this proposed patent quality classifier to other classification models, the experimental findings show that it has the best classification performance [23].

Recently, LSTM is also used in any studies. LSTM stands for "long short-term memory." It is a recurrent neural network (RNN) made to assess time series or spoken language using sequential data. Because of their ability to retain and retrieve information over long periods of time. For tasks where the model must remember and use long-term dependencies, LSTMs are especially successful. In the context of patent analysis, LSTMs may be used to analyze the content of patent documents, such as the descriptions of inventions and the claims made by the inventors [17].

This could be used to extract relevant information from the patents, classify the patents according to their subject matter, or identify trends in the technological field that the patents pertain to. For example, a company might use an LSTM model to analyze a large dataset of patents in order to identify new technologies that could be of interest to the company, or to better understand the competitive landscape in a particular market.

2.6.4 Graph Neural Networks (GNNs)

GNNs (Graph Neural Networks) are deep learning models that act on graph-structured data. In the context of patent analysis, GNNs can be used to analyze the relationships between patents and their citations, as well as the relationships between different patents and the companies or individuals who hold them. To understand how GNNs work, it is helpful to first understand the concept of a graph. A graph is a data structure comprising nodes (also known as vertices) and edges. Nodes symbolize the entities under analysis, while edges represent the relationships that exist between these entities.

For example, in the context of patent analysis, the nodes might represent patents and the edges might represent citations between those patents. GNNs are trained to operate on graph-structured data by passing messages between the nodes in the graph. The messages that are passed between nodes are called "node embeddings," and they represent the characteristics or features of the nodes. The node embeddings are then used to make predictions or decisions based on the graph structure and the relationships between the nodes.

In the context of patent analysis, GNNs can be used to identify patterns and trends in the relationships between patents and their citations, as well as to predict which patents are likely to be most influential or valuable. GNNs can also be used to identify relationships between patents and the companies or individuals who hold them, which can be useful for identifying patterns of innovation and collaboration within the patent landscape [32].

With the aid of the technological association and the proposed patent ontology, a study constructs a patent heterogeneous network. Ultimately, to leverage the technological features of patents fully, a heterogeneous graph embedding technique is constructed to integrate this piece of information into the patent representation. Through experiments conducted on non-perfluorinated proton exchange membrane patent-related data, it is evident that our approach surpasses similar models concerning patent representation performance[43]. Overall, GNNs offer a powerful tool for analyzing and understanding the complex relationships within large sets of patent data, and they have the potential to revolutionize the way that patent analysis is performed.

Deep learning architectures in use now are more diversified. The more complicated network designs are gaining popularity, although the more traditional network architectures, CNN (Convolutional Neural Networks) and LSTM (Long Short-Term Memory) remain the most prevalent choices for conducting research in patent analysis. The primary task in patent analysis is classification, which benefits from the fact that all published patents are assigned specific classes. This results in a substantial amount of labeled training data available in its most basic form, making classification evaluation relatively straightforward. Furthermore, patent retrieval and its related tasks are highly favored among researchers in this field.

Only a minimal amount of research has been done so far on the remaining highlighted jobs. The fact that these jobs are more challenging is one explanation for this. Automatic approaches still struggle with problems where even human specialists cannot agree on a solution or where the work necessitates a great deal of common sense and prior knowledge. However, we anticipate that in the future, more powerful deep-learning techniques will be better equipped to manage these challenging jobs.

2.7 Models employed in Study

To enhance the performance of the foundational models in Patent Sentiment Analysis, we conducted training on five distinct deep learning-based models using the Patent dataset. These models include Simple Feed-forward Neural Network, Long Short-Term Memory (LSTM), BERT, RoBERTa, and DistilBERT.

2.7.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is an NLP model developed by Google. It has gained significant attention for its ability to understand contextual language representations bidirectionally, which helps it excel in various natural language processing tasks. It is intended to preprocess and comprehend the context of words in a phrase by taking into account the words that come before and after them. BERT is a transformer-based architecture that processes input sequences using selfattention methods. It is trained on a large dataset of unlabeled text and a small dataset of labeled text and is able to effectively capture the meaning and context of words in a sentence by considering the words around them [36].

BERT has been utilized to deliver cutting-edge outcomes in a wide range of NLP applications, including question-answering, language translation, and sentiment analysis. It has also been used to improve the performance of other NLP models by fine-tuning them on specific tasks. One of the tasks that BERT can be used for is patent analysis. In the context of patent analysis, BERT can be used to process and analyze the text of patent documents [19]. This can involve tasks such as extracting information from the patent text, identifying relevant patents for a given query, and generating summaries of patent documents.

To perform patent analysis using BERT, the model would first be fed the text of a patent document or a set of patent documents. The model would then process the text, using its knowledge of language and its ability to understand the context and relationships between words and phrases in the text. This can involve tasks such as identifying named entities (such as the names of inventors or companies), extracting technical terms and concepts, and understanding the meaning and importance of specific words and phrases within the context of the patent document.

Once BERT has processed the text of the patent documents, it can be used to perform a range of analysis tasks, such as identifying relevant patents for a given query, generating summaries of patent documents, or extracting specific information from the patent text. This can be useful for tasks such as patent search and retrieval, patent portfolio management, and patent analysis for technology scouting and competitive intelligence.

2.7.2 Robustly Optimized BERT Approach (RoBerta)

RoBerta (short for "Robustly Optimised BERT Approach") is a language model based on BERT (Bidirectional Encoder Representations from Transformers). BERT, a neural network architecture founded on transformers, underwent extensive training to excel in a diverse array of natural language processing tasks, encompassing language translation, text classification, and language synthesis. RoBerta is specifically designed to be a more robust version of BERT, with improved generalization capabilities and reduced overfitting to the training data. It was created by Facebook AI researchers and has demonstrated cutting-edge performance on a variety of natural language processing benchmarks. RoBerta can be used for patent analysis in a number of ways, depending on the specific goals of the analysis and the data available. Some possible applications of RoBerta for patent analysis include:

- Patent classification: RoBerta can be trained to classify patent documents into different categories or classes based on the content of the patent. For example, RoBerta could be used to classify patents based on the technology they relate to (e.g., pharmaceuticals, electronics, mechanical devices).
- Patent search: RoBerta could be used to improve the accuracy and relevance of search results for patent databases. By training RoBerta on a large dataset of patent documents and associated metadata (e.g., title, abstract, claims), it can learn to understand the content and context of the patents and match search queries to relevant patents more accurately.
- Patent summarization: RoBerta could be used to generate concise summaries of patent documents, which could be useful for quickly understanding the key points of a patent or for creating a summary of a large number of patents for review or analysis.
- Patent translation: RoBerta could be used to translate patent documents from one language to another, which could be useful for making patent information more accessible to a wider audience or for comparing patents in different languages.
- Patent trend analysis: RoBerta could be used to analyze trends in patent activity over time, such as the number of patents filed in a particular technology area or the countries where patents are being filed. This could be useful for identifying emerging technologies or for tracking the patent activity of a particular company or group of companies [29].

Automating Patent Analysis using RoBerta RoBerta can be used to automate various aspects of patent analysis, which can save time and effort compared to manually analyzing patents. Some specific ways in which RoBerta can be used to automate patent analysis include:

- Automating patent classification: By training RoBerta on a large dataset of patent documents and their associated class labels, it can learn to classify new patent documents accurately and automatically. This can save time compared to manually reviewing and classifying each patent.
- Automating patent search: RoBerta can be used to improve the accuracy and relevance of search results for patent databases, which can save time and effort compared to manually reviewing search results.
- Automating patent summarization: RoBerta can be used to generate concise summaries of patent documents, which can save time and effort compared to manually reading and summarizing each patent.
- Automating patent translation: RoBerta can be used to translate patent documents from one language to another, which can save time and effort compared to manually translating each patent.

Overall, the use of RoBerta for automating patent analysis can help to improve the efficiency and effectiveness of the analysis process, as well as enabling the analysis of larger volumes of patent data.

2.7.3 DistilBERT

DistilBERT, a compact and quicker rendition of the BERT language model (Bidirectional Encoder Representations from Transformers), offers enhanced efficiency. BERT, founded on transformers, was trained to excel in numerous natural language processing steps, inclusive of language translation, text classification, and language synthesis. DistilBERT was developed by researchers at Hugging Face as a way to make BERT more accessible and easier to use for a wider range of applications.

It was created by distilling, or compressing, the knowledge of a larger BERT model into a smaller, more efficient model. DistilBERT is approximately 40% the size of BERT and can be trained up to 6 times faster, while still maintaining a similar level of performance on a range of natural language processing tasks[44]. It has been shown to perform almost as well as BERT on a number of benchmarks and is widely used in a variety of natural language processing applications.

DistilBERT can be used to automate various aspects of patent analysis, just like RoBerta. Some specific ways in which DistilBERT can be used to automate patent analysis include:

- Automating patent classification: By training DistilBERT on a large dataset of patent documents and their associated class labels, it can learn to classify new patent documents accurately and automatically. This can save time compared to manually reviewing and classifying each patent.
- Automating patent search: DistilBERT can be used to improve the accuracy and relevance of search results for patent databases, which can save time and effort compared to manually reviewing search results.
- Automating patent summarization: DistilBERT can be used to generate concise summaries of patent documents, which can save time and effort compared to manually reading and summarizing each patent.
- Automating patent translation: DistilBERT can be used to translate patent documents from one language to another, which can save time and effort compared to manually translating each patent.

Overall, the use of DistilBERT for automating patent analysis can help to improve the efficiency and effectiveness of the analysis process, as well as enabling the analysis of larger volumes of patent data[41].

Chapter 3

Methodology

This section of the thesis is for automating patent analysis employing deep learning models.

3.1 General Architecture

We offer a generic architecture that depicts each stage of our suggested research framework. Our suggested framework begins with data collection and ends with patent classification.

3.2 Dataset Acquisition

The United States Patent & Trademark Office (USPTO) raw XML files were used to create a curated, selectively extracted collection known as the "patent sentiment analysis dataset." To encourage innovation and creativity, the USPTO provides the full text of patent grants in nested XML structured files publically. Each year, grants are published and stored weekly in zipped files (for eg: ipg210107.zip, first week of January 2021). The link, for instance, contains 52 XML files that are organized in accordance with each week of the year 2020.

There are 52 nested XML files, each of which contains all grants that were published during that week. These XML files were then parsed and data is gathered in a CSV file. The dataset contains 150,000 labeled examples. We split the dataset into 80:20 train/test split randomly. Train-set contains 120,000 instances while 30,000 instances were used as test-set. Train/test sets were stored independently so that train/test sets remains the same for every experiment.

3.3 Models

In this thesis, we have used some Deep Learning models for our experimentation including FFNN, LSTM, BERT, DistilBERT, Roberta.

3.3.1 FFNN

A recurrent neural network (RNN) is referred to as a type of artificial neural network with a network of nodes, similar to a feed-forward neural network. However, RNNs differ from feed-forward networks because they include cycles or loops in some paths. In contrast, feed-forward neural networks do not have any cycles, making them process input data in a unidirectional manner. This unidirectional processing of input data makes the feed-forward model one of the fundamental types of neural networks. Artificial neural networks that do not feature looping nodes are called feed-forward networks. This kind of neural network is also referred to as a multi-layer neural network because all input is simply transferred forward. Receiving data at input nodes, sending it across unobservable layers, and outputting it at nodes are all components of data flow. The network has no manipulatable links that may be used to send data back from the output node[9].

The following is how a feed-forward neural network approximates functions:

1. The formula y = f*(x) is used in an algorithm to determine classifiers.
2. As a result, input x is categorized under y.
3. The feed forward model states that y = f (x; 0). The closest approximation of the function is determined by this value.

The architecture diagram for the FFNN model is shown below in Figure 3.1. When simplified, the feed-forward neural network could seem like a single-layer perceptron. Weights are multiplied by inputs as they enter the layer in this model. The sum is then calculated using the weighted input values. The output result is normally 1 if the

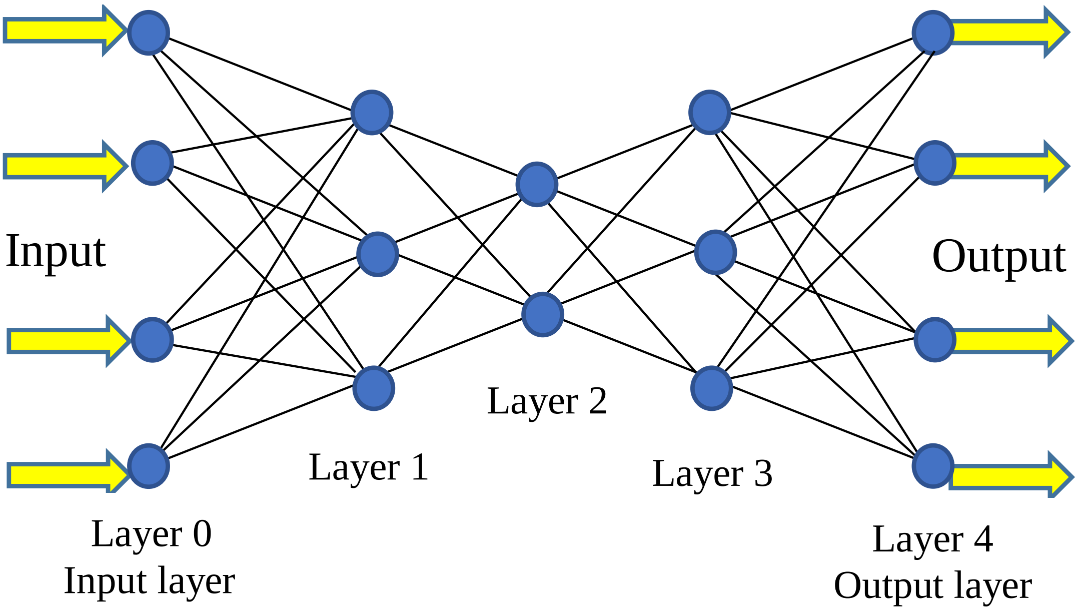


Figure 3.1: FFNN model

total of the values exceeds a predefined threshold, which is set at zero; otherwise, it is typically -1. It has three layers that are:

- Input Layer: Neurons within this layer receive information and transmit it to the other layers of the network. It is essential that the number of neurons in the input layer corresponds to the feature or attribute counts present in the dataset.
- Hidden Layer: The input and output layers are situated between hidden layers. The model may contain multiple hidden layers, each consisting of numerous neurons that modify the input before passing it to the subsequent layer. The network's weights are continuously adjusted to enhance predictability and improve the overall performance of the model.
- Output Layer: This layer represents the projected feature, which varies depending on the type of model being built.
- Activation Function: Making decisions in this area is the responsibility of neurons. The neurons choose whether to make a linear or nonlinear judgment based on the activation function. It avoids the cascade effect by moving through so many levels, which keeps neuron outputs from rising. There are three main groups of activation functions: sigmoid (the output values are translated to input values between 0 and 1), Tanh (the input data are translated into a value between -1 and 1), and

Rectified Linear Unit (ReLu that can only pass over positive values. Negative values are assigned to a value of 0).

- Cost Function: The cost function is fundamental in a feed-forward neural network. Minor weight and bias changes have little impact on the categorized data points. Thus, a strategy of modifying weights and biases to increase performance can be determined using a smooth cost function. A description of the mean square error cost function is illustrated by Equation 3.3.1.

  $$C(w, b) \equiv \frac{1}{2n} \sum_{x} \|y(x) - a)\|^2 \tag{3.3.1}$$

  $Where$

  w = the weights gathered in the network

  b = biases

  n = number of inputs for training

  a = output vectors

  x = input

  $\|v\|$ = vector v's normal length

- Loss Function: The loss function of a neural network serves as a means to evaluate whether adjustments are necessary during the learning process. It quantifies the discrepancy between the expected and actual probability distributions, which becomes apparent when the number of classes and neurons in the output layer are equal. Equation 3.3.2 illustrates the cross-entropy loss in the binary classification.

  $$Loss = -\frac{1}{k} \sum_{i=1}^{k} y_i . \log \hat{y}_i + (1 - y_i) . \log(1 - \hat{y}_i) \tag{3.3.2}$$

  $Where$

  k = output size is the number of scalar values in the model output.

  A cross-entropy loss results from multi-class categorization with Equation 3.3.3.

  $$Loss = -\sum_{i=1}^{k} y_i . \log \hat{y}_i \tag{3.3.3}$$

  With a moderated intermediary, many networks in feed-forward networks operate independently. Multiple neurons are needed in the network for complex tasks.

3.3.2 LSTM

LSTM stands for Long Short-Term Memory, a specialized type of recurrent neural network (RNN) recognized for its capacity to proficiently learn long-term dependencies in sequential data. The concept of gates is used by LSTMs to simplify and successfully conduct calculations using both Long Term Memory (LTM) and Short Term Memory (STM).

- Forget Gate: Unhelpful information is forgotten by LTM as it passes through the forget gate. Event (Et) and Prior Short-Term Memory (STMt-1) are the inputs, and only significant data is kept for prediction. The calculation for this gate is below. The weight matrix Wn is used in conjunction with the tanh (hyperbolic Tangent) function to introduce nonlinearity, resulting in the matrix Nt. The previous short-term memory vector STMt-1 with the current event vector Et are concatenated as [STMt-1, Et] and then multiplied with bias in the weight matrix. This fusion of the present event vector Et and short-term memory STMt-1 generates a single ignore factor, which is subsequently multiplied by the weight matrix Wi and processed through a sigmoid activation function with bias.

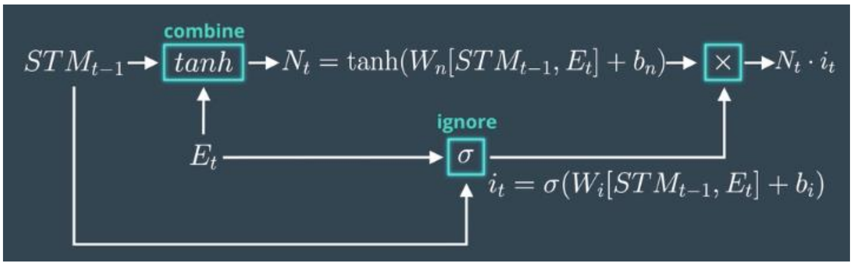

combine
STM_{t-1}
tanh
$N_t = \tanh(W_n[STM_{t-1}, E_t] + b_n)$
×
$N_t \cdot i_t$
ignore
E_t
σ
$i_t = \sigma(W_i[STM_{t-1}, E_t] + b_i)$

Figure 3.2: Forget Gate

To obtain the learn gate output, you need to multiply the learning matrix Nt by the ignore factor. It is illustrated in the Figure 3.2.

- Learn Gate: By combining the event (current input) and STM, it is possible to subject the current input to the most recent knowledge discovered by STM. Which knowledge should be retained and which should be forgotten is decided using data from the Previous Long Term Memory (LTMt-1). The calculation for this gate is below in Figure 3.3.

The Forget Factor ft is calculated by combining the last Short Term Memory (STMt-1) and the present Event vector (Et) as [STMt-1, Et]. Next, this combined vector is multiplied by the weight matrix Wf, and the results are passed through the Sigmoid activation function with the inclusion of specific bias. The output of the forget gate is produced by multiplying the Forget Factor ft with the Previous LTMt-1.

- Remember Gate: Remember Gate functions as an updated LTM by integrating LTM data that we haven't forgotten with STM and Event data. Combine Previous Short Term Memory (STMt-1) with Current Event (Et) to create output. The calculation for this gate is below in Figure 3.4. The output is obtained by combining the past STMt-1 with the Current Event (Et). The Remember Gate generates the Long Term Memory (LTM) for the subsequent cell by combining the outputs from both the Forget Gate and the Learn Gate.

- Use Gate: This gate utilizes the Long Term Memory (LTM), Short Term Memory (STM), and Current Event to predict the future outcome of the current event, effectively acting as an updated STM. To generate Short Term Memory (STM) for the subsequent cell and produce output for the current event, the combination of both Short Term Memory and Long Term Memory from previous steps is used.

  The Prior Long Term Memory (LTM-1) undergoes the bias-added Tangent activation function, resulting in Ut. Next, the Current Event (Et) and Prior Short-Term Memory (STMt-1) are combined to create Vt. The output of the Use Gate, which also serves as the Short-Term Memory (STM) for the next cell, is generated by multiplying Ut and Vt together. It is illustrated in Figure 3.5.

3.3.3 BERT

One of the most significant challenges in NLP is insufficient training data. While there is a substantial amount of text data available, it is essential to divide it into various fields to create task-specific datasets. As a result, the remaining training instances typically

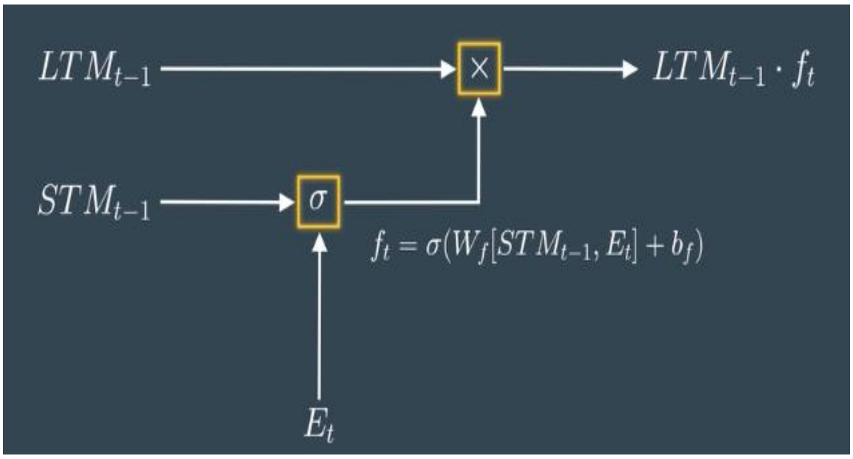


Figure 3.3: Learn Gate

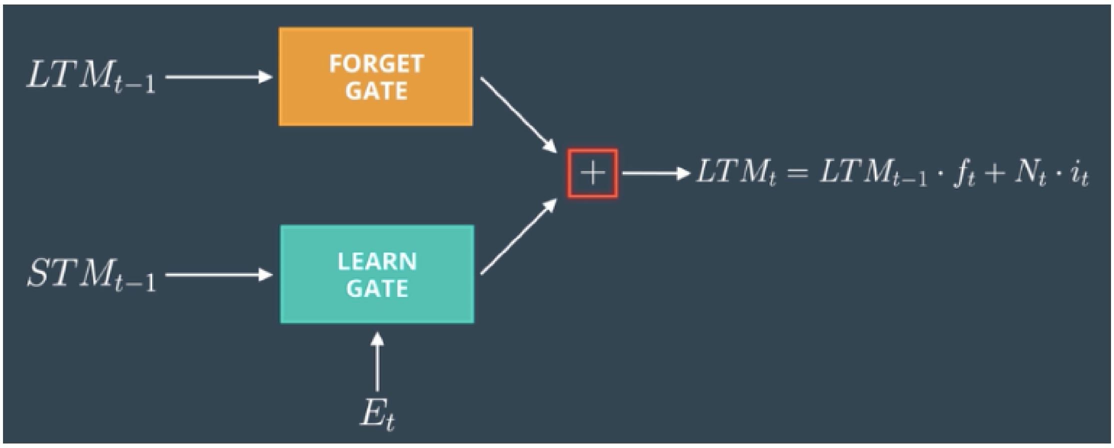


Figure 3.4: Remember Gate

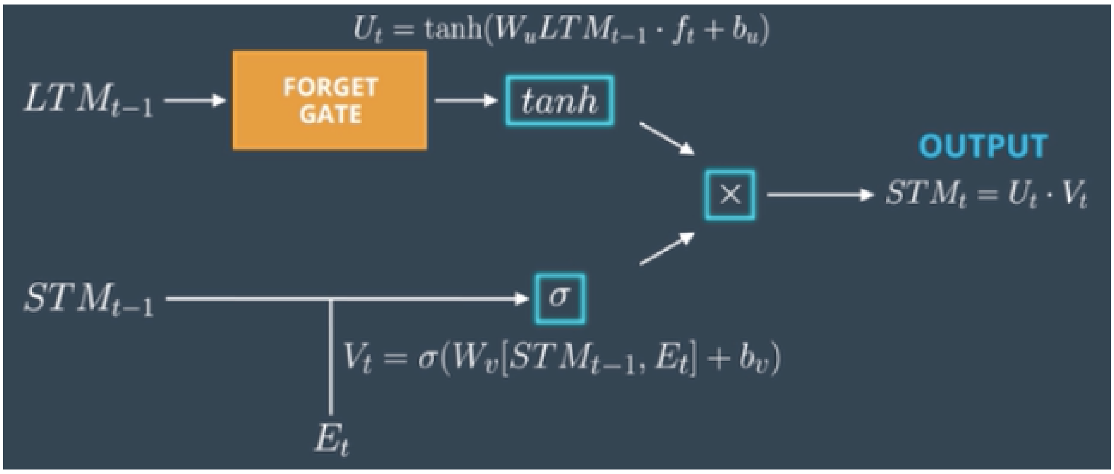

$U_t = \tanh(W_u LTM_{t-1} \cdot f_t + b_u)$
LTM_{t-1}
FORGET GATE
$tanh$
×
OUTPUT
$STM_t = U_t \cdot V_t$
STM_{t-1}
σ
$V_t = \sigma(W_v[STM_{t-1}, E_t] + b_v)$
E_t

Figure 3.5: Use Gate

range from a few thousand to a few hundred thousand, and they need to be manually labelled by humans.

Indeed, deep learning-based NLP models require a substantial amount of data to perform optimally. Provided millions or billions of annotated training examples, these models show significant improvement in their performance and capabilities. The availability of large-scale annotated datasets is crucial for enhancing the accuracy and effectiveness of deep learning-based NLP models[38]. To address this data gap, Several techniques have been developed for the pre-training of general-purpose models for language representation using extensive volumes of unannotated web data. These pre-training methods enable NLP models to learn from a vast amount of unlabeled text, which helps in improving their performance on downstream tasks with limited labelled data.

Furthermore, while countering tasks such as question-answering and sentiment analysis, intensively trained general-purpose When tackling tasks like question answering and sentiment analysis, these general-purpose pre-trained models can be further fine-tuned using smaller task-specific datasets.

This approach leads to a significant boost in accuracy compared to training from scratch using limited task-specific data. BERT, a relatively recent addition to NLP pre-training methodologies, garnered attention in the deep learning community due to its exceptional performance across various NLP tasks, including question answering, setting new benchmarks in the field [33].

BERT is built upon the Transformer model architecture, which operates in a series of small, regular steps. At each step, the model utilizes an attention mechanism to establish connections between all words in a sentence, regardless of their positions. The basic structure of a Transformer comprises an encoder responsible for reading the input text and a decoder responsible for generating task predictions. Since BERT aims to create a language representation model, just the encoder part is employed. The encoder's input is a series of tokens that are transformed into vectors and are computed by the neural network. However, before processing begins, BERT requires additional metadata to update and enhance the input.

- Token Embeddings: The input word tokens are augmented with a [CLS] token at the start of the first sentence and a [SEP] token at the end of each and every sentence. These special tokens play essential roles in BERT's architecture and enable the model to understand the sentence boundaries and distinguish between different sentences in the input.

- Segment embeddings: Sentence A or Sentence B are written on each token, respectively. Because of this, the encoder may detect different sentences.

- Positional embeddings: To indicate the position of each token in the text, a positional embedding is assigned to it. This positional embedding provides crucial information to the model, allowing it to understand the relative positions of words in the input sequence and consider the context in which each word appears. By incorporating positional embeddings, BERT can effectively capture the sequential information and comprehend the relationships between different words in the input text.

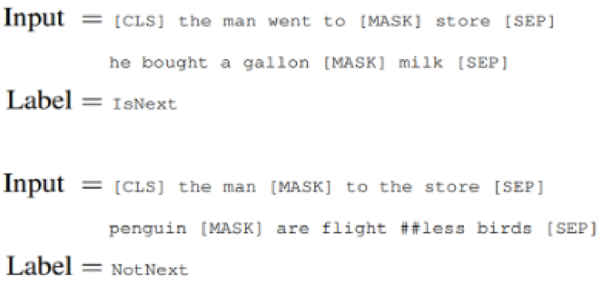
  In summary, the Transformer model consists of stacked layers that map sequences to sequences, yielding an output that is also a sequence of vectors. Each input and output token corresponds one-to-one at the same index. Notably, BERT takes a different approach and does not aim to forecast the following word in the sentence. Instead, it employs a masked language modelling objective, where certain tokens are masked in the input, and the model predicts those masked tokens based on the context of the surrounding words. Two strategies are employed during training:

  - Masked LM (MLM)

    15% of the input words should be randomly chosen for masking and [MASK] token replacement. The main idea is as follows: After the complete sequence is processed through the BERT attention-based encoder, the model focuses solely on predicting the masked words. It is accomplished by employing the context given by the other non-masked phrases within the sequence. This approach enables BERT to learn contextually relevant representations for the masked words, improving its performance in various natural language processing tasks.

  - Next Sentence Prediction (NSP)

    During training, when provided with pairs of sentences as input, the BERT model learns to predict whether the second sentence appears after the first one in the original text. This ability to understand sentence order is particularly useful for tasks like question-answering. As we know, BERT uses the [SEP]



```
Input = [CLS] the man went to [MASK] store [SEP]
        he bought a gallon [MASK] milk [SEP]
Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]
        penguin [MASK] are flight ##less birds [SEP]
Label = NotNext
```

token to separate sentences. In the training process, the model accepts two input sentences simultaneously, with the second sentence being the actual continuation of the first one 50% of the time.

The leftover 50% of the time, a random statement from the corpus is used as the second sentence. In such cases, BERT must determine if the random statement is connected to the first sentence to ascertain if the second sentence is random or it is not. This enables BERT to acquire contextual understanding and improve its performance in a wide range of language-related tasks. In essence, the Transformer-based model processes the entire input sequence to establish whether the second sentence is connected to the first. A simple classification layer is employed to convert the output of the [CLS] token into a vector with 21 dimensions, and softmax is utilized to determine the IsNext-Label. The model is trained by combining MLM (Masked Language Modeling) and NSP (Next Sentence Prediction). The combined loss function of these two strategies is minimized during training, allowing the model to effectively learn the contextual representations and improve its performance in sentence-pair tasks [26].

The architecture diagram for the BERT model is shown in Figure 3.6.

3.3.4 DistilBERT

In this study, DistilBERT and BERT share the same fundamental architecture. However, in DistilBERT, the layer count is reduced by half, and the token- and pooler-type embeddings are removed. Our research findings indicate that variations in factors like the number of layers have a more significant impact on computation efficiency compared to changes in the tensor's last dimension (hidden size dimension) for a given parameter

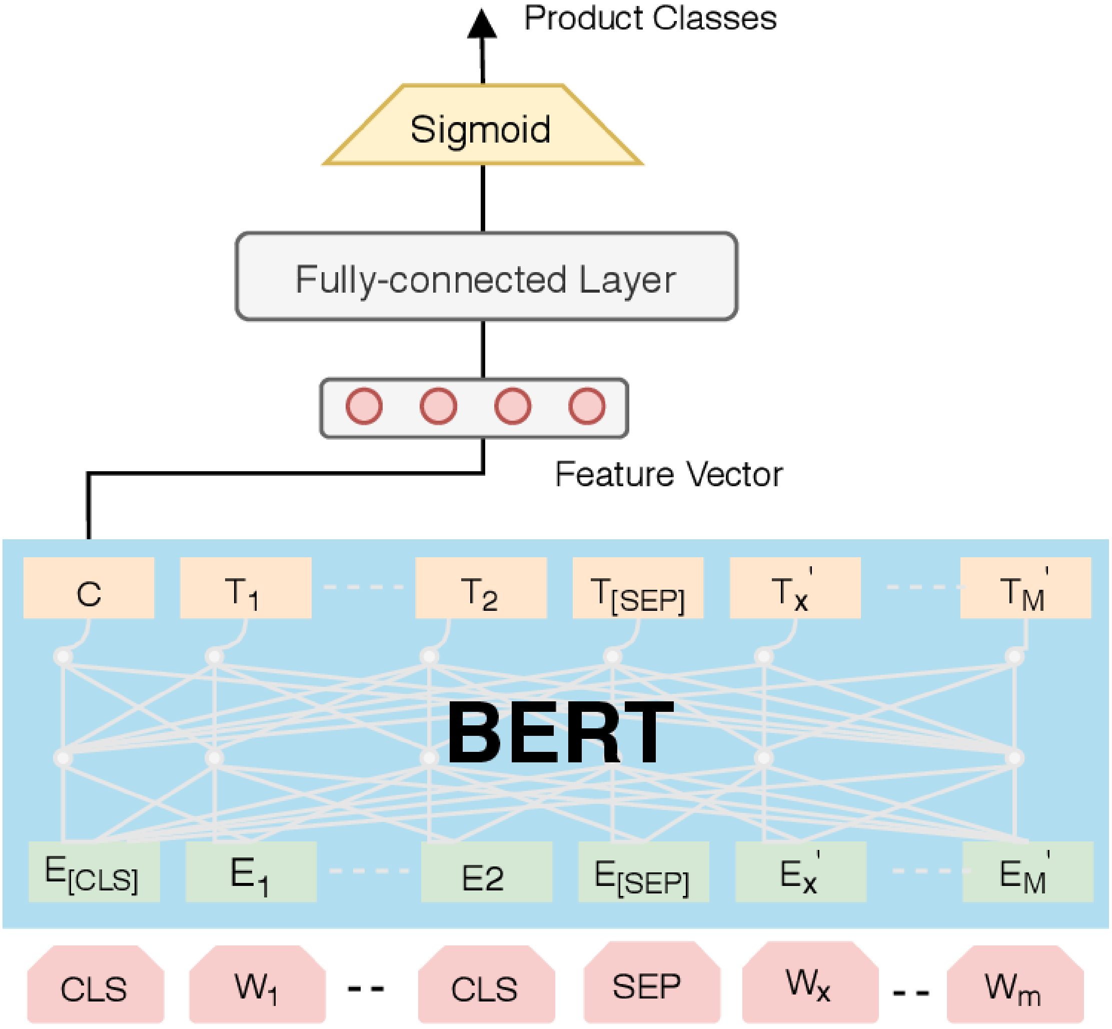


Figure 3.6: BERT Architecture

budget. Most of the processes used in the Transformer design, such as linear layers and layer normalization, are highly optimized in modern linear algebra frameworks. As a result, our primary focus is on removing layers to improve computation efficiency[21].

Apart from the optimization and architecture options discussed earlier, another crucial stage in our training approach is finding the optimal initialization for the sub-network to converge effectively. We initiate the student network from the teacher network by taking out one layer out of every two, leveraging the dimensionality shared between the teacher and student network graph. This initialization strategy aids in facilitating the training process and improving the overall performance of the student network[21].

3.3.5 roBERTa

The Optimised (robustly) BERT Pre-training Approach, or RoBERTa, was developed by scientists from Washington University and Facebook. Its main objective is to speed up and improve the pre-training of the BERT architecture. Although RoBERTa's design is closely akin to the BERT, the authors done a number of small adaptations to the training phase as well as architecture to outperform the original BERT model[28]. These customizations are:

- Removing the NSP objective:
  In the next sentence forecasting task, the trained model is to identify whether the observed document segments are from the same or separate texts using an auxiliary Next Sentence Prediction (NSP) loss. The researchers evaluated several versions with and without NSP loss and found that doing so matches or slightly improves downstream task performance.

- Bigger batch sizes & longer sequences based training:
  Originally BERT is trained for 256-sequence batches with over 1M steps. This work employed 125 steps with each 2,000 sequences having 31,000 steps of batch size 8,000 sequences to train the model. Larger batch sizes have two advantages: they improve end-task accuracy and confusion on the masked language modelling objective. Large batch parallelization is also made simpler by distributed parallel training.

- Changing the masking pattern dynamically

BERT architecture, masking is done only once, during data preparation, generating a single static mask. Training data is repeated 10 times using different masking techniques throughout the period of 40 epochs, avoiding the use of a single static mask. Four epochs with the same mask are the result. The dynamic masking technique, which produces a new mask each time input is provided to the model, is not used in this method.

It generates better outcomes than the BERT(LARGE) model [34].

Chapter 4

Experimentation, Results and Discussion

This chapter contains information on the environmental setup for assessing and classifying patents, experimentation information, and experiment outcomes, such as the amount of time needed to train the model and the accuracy matrices of the findings.

4.1 Environmental Setup

The implementation is done using Python for experimenting on the Google Colaboratory Environment (Colab) with the Graphics Processing Unit (GPU) activated. There are various third-party libraries available for data profiling, pre-processing, error reporting, visualization, model training, and creating result metrics. Pandas, Seaborn, Sklearn, Numpy, Torch, Tqdm, Transformers, and Logging are among them.

4.1.1 Google Colaboratory

The Google Colaboratory is a cloud-based environment integrated development environment (IDE) that utilizes Jupyter notebooks that are hosted in the cloud and are tightly integrated with Google Drive, making it simpler to use, share, and work together on projects with other researchers and developers. As a result, the drive integration completely eliminates the system dependency. Researchers and developers can store their data on Drive and use Colab from any location. The IDE has zero configuration requirements, works with thousands of third-party libraries, offers markdown cells for describing experiment phases, and can run several sessions simultaneously on CPUs, GPUs, and Tensor Processing Units (TPUs).

4.2 Experimentation

We performed our experiments on the free version of Google Colab. Google Colab provided us with 12GB of RAM memory and a T4 GPU with 16GB of VRAM memory. The dataset contains 150,000 labelled examples. We split the dataset into 80:20 train/test split randomly. Train-set contains 120,000 instances while 30,000 instances were used as test-set. Train/test sets were stored independently so that train/test sets remains the same for every experiment.

In order to improve the performance of the base models of Patent Sentiment Analysis, we trained five different deep learned-based models on the Patent dataset: Simple Feedforward Neural Network, Long Short-Term Memory (LSTM), BERT, RoBERTa, and DistilBERT.

We have trained different FeedForward Neural Networks (FFNN) with different hyperparameters and logged our precision, recall, and f1-score in Table 4.1 and confusion matrix in 4.2. We used pre-trained GloVe embeddings as word embeddings. Sentence embeddings were calculated as the element-wise mean of all word vectors of that sentence.

4.2.1 Training FFNN

We have trained FFNN for 500 epochs and logged the loss and test the accuracy of the model after each epoch. The loss and accuracy graphs are smoothed using exponential moving averages so we can see the overall trend of the model. It can be seen in Figures 4.1 and 4.2 respectively. The original values of accuracy and loss can also be seen on the graph. The model gives the best accuracy on the 22nd epoch. After that 22nd epoch, the test accuracy slowly starts decreasing.

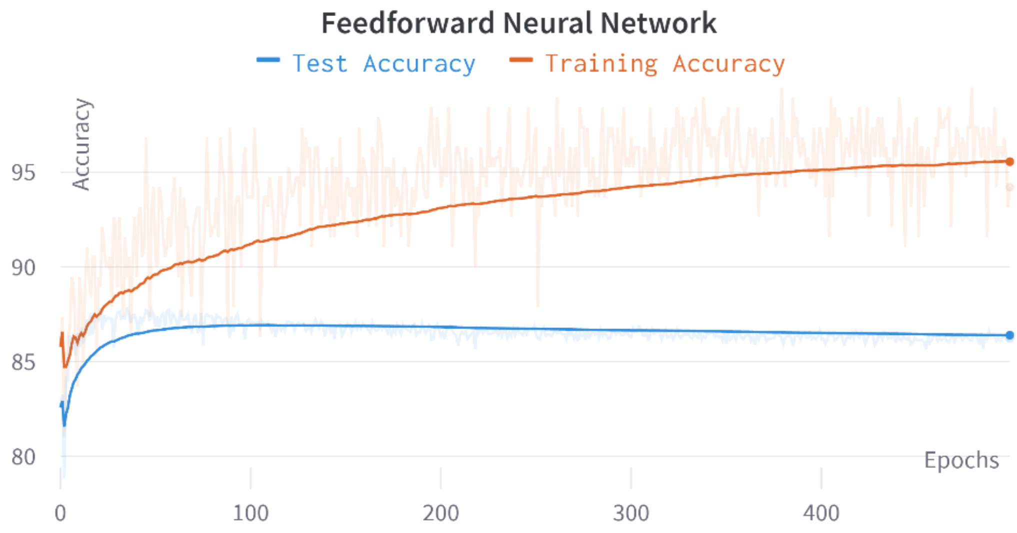


Figure 4.1: FFNN: Train and Test accuracy during 500 Epochs

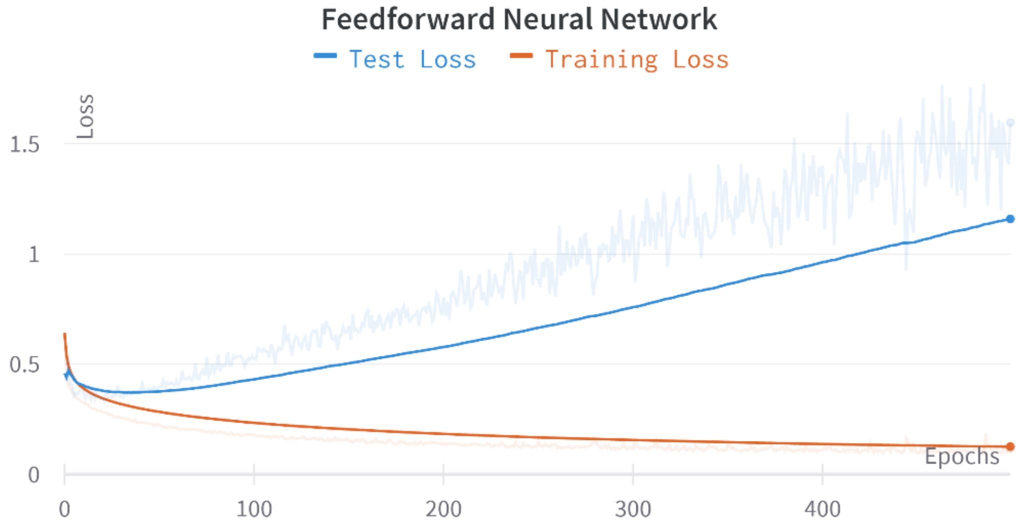


Figure 4.2: FFNN: Train and Test loss during 500 Epochs

We used Dropout with a probability of 0.5 before the final layer of the network so that the model does not overfit in early epochs. The test loss starts decreasing after the 22nd epoch, which indicates that the model has started overfitting the data. We showed the results of the 22nd epoch of FFNN in this paper: see Table 4.1. FFNN achieves an overall accuracy of 87.83% and an f-1 score of 0.88. The model achieves the best f-1 score of 0.90 for class 0. Confusion Matrix is also shown in Table 4.2 to get better insights into the model's predictions.

4.2.2 Training LSTM

We have trained a simple LSTM model for 50 epochs and logged the training and testing accuracy and loss at each epoch. See Figures 4.3 and 4.8. We evaluated the model on test-set after each epoch. It turned out that the 14th epoch gives us the best test accuracy of 97.32% and an f1-score of 0.97. The class-wise precision, recall, and f1-score is given in Table 4.3 and the confusion matrix is in Table 4.4. All three classes get an f1-score of 0.97.

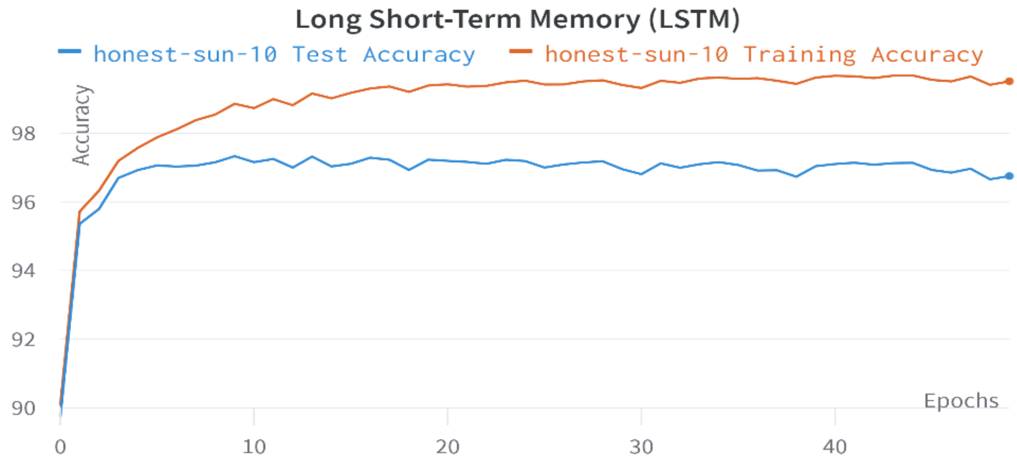


Figure 4.3: LSTM: Train and Test accuracy during 50 Epochs

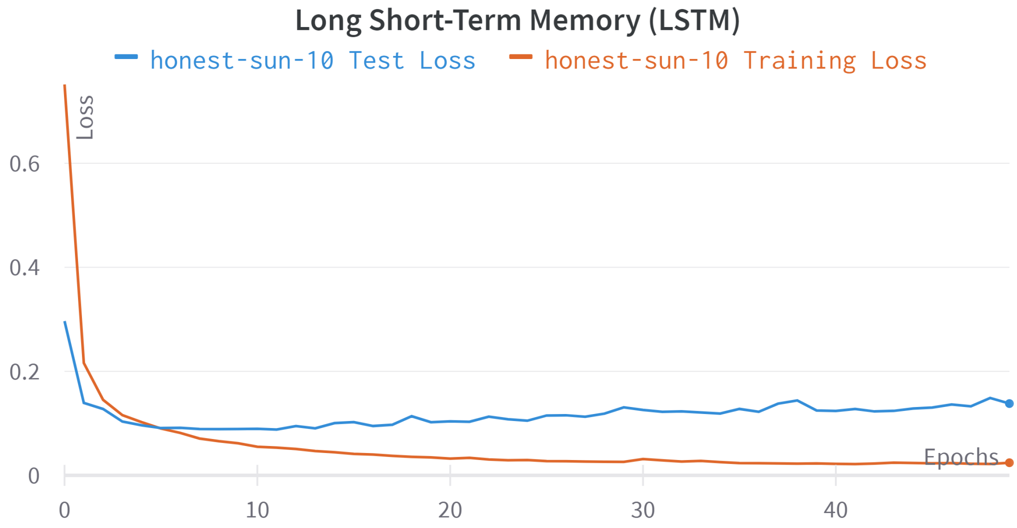


Figure 4.4: Train and Test loss during 50 Epochs

4.2.3 Training BERT, Roberta and DistilBERT

We have fine-tuned three pre-trained transformer-based models BERT, RoBERTa, and DistilBERT each for 1 epoch. These are large models so we kept the batch size to 16 to fit it into memory. All three models converge almost at the same rate: see Figure 4.5. The graphs are smoothed by exponential moving averages so that we can easily notice the trend of the loss. After 1 epoch, all three models achieve an f1-score of 0.98. DistilBERT and BERT have an accuracy of 98.17% while the RoBERTa achieves slightly better accuracy of 98.39%. Considering individual classes, BERT and DistilBERT achieve an f1-score of 0.98 for all three classes while RoBERTa achieves the best f1-score of 0.99 for class 0. The precision, recall, and f-score for BERT, RoBERTa, and DistilBERT are given in Table 4.7, 4.9, and 4.5 respectively. The confusion matrices are given in Tables 4.8, 4.10, and 4.6 respectively.

4.3 Results

We compared the performance of our five different models in Table 4.11. It turned out that RoBERTa performs best in terms of precision, recall, f1-score, and accuracy. We also compared training time per epoch, the total number of trainable parameters, the model size on disk, the total number of training epochs, batch size, and the Maximum sequence length of the models for each model. We have noticed that even though the RoBERTa performs better than all other models in terms of accuracy but it has more number of parameter which increases the training time of the model.

Hence one epoch takes 3 hours, 26 minutes, and 4 seconds to train. We also noticed that BERT and DistilBERT have the same performance but DistilBERT is 1.67 times smaller than BERT and has a smaller number of parameters and disk size. DistilBERT trains faster than BERT and RoBERTa. Finally, we compared the performance of our models with the base models. LSTM, BERT, RoBERTa, and DistilBERT perform better than all base models by some margin. The Feedforward Neural Network performs better than the RFC model but it cannot beat other base models.

Based on our study and literature, we discovered that different regions of the world have different patent drafting procedures and formats. For instance, in order to maximize their chances of being granted a patent, applicants from Asian countries tend to explicitly list several benefits of their innovation and disadvantages related to the sources they cited. Patents with such clearly defined structures as specific sections make it easier for examiners and attorneys to swiftly go through documents to find important highlights and debatable subject topics.

Resultantly, we looked up such unique tags that could serve as important entities amidst the inspection and comparison of any current literature. As seen in the Figure above, three potential tags are chosen, however, they are not common across all patents. In other words, the availability of such tags is frequently dispersed. For instance, for the year 2020, only about 8000 patents were discovered with such explicit text segments out of almost 200000 grants. In order to facilitate patent analysis, it is important to create a dataset having special text segments and models trained to dynamically highlight the text in a given patent that lack particular tags.

Obsserve Figure 4.5 and 4.6. The loss and accuracy graphs are smoothed using exponential moving averages so we can see the overall trend of the model. The original values of accuracy and loss can also be seen on the graph. The model gives the best accuracy on the 22nd epoch. After that 22nd epoch, the test accuracy slowly starts decreasing.

Figure 4.5: FFNN: Train and Test accuracy during 500 Epochs

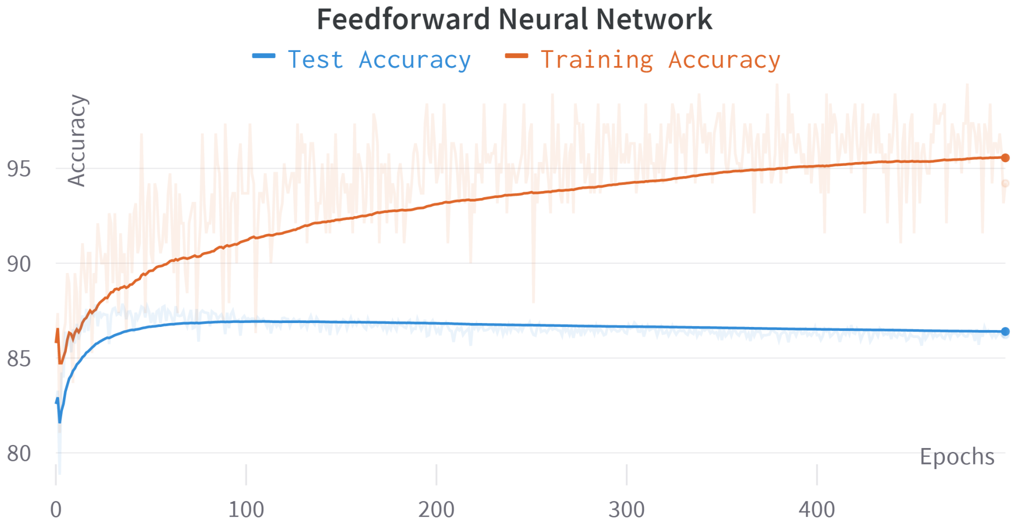
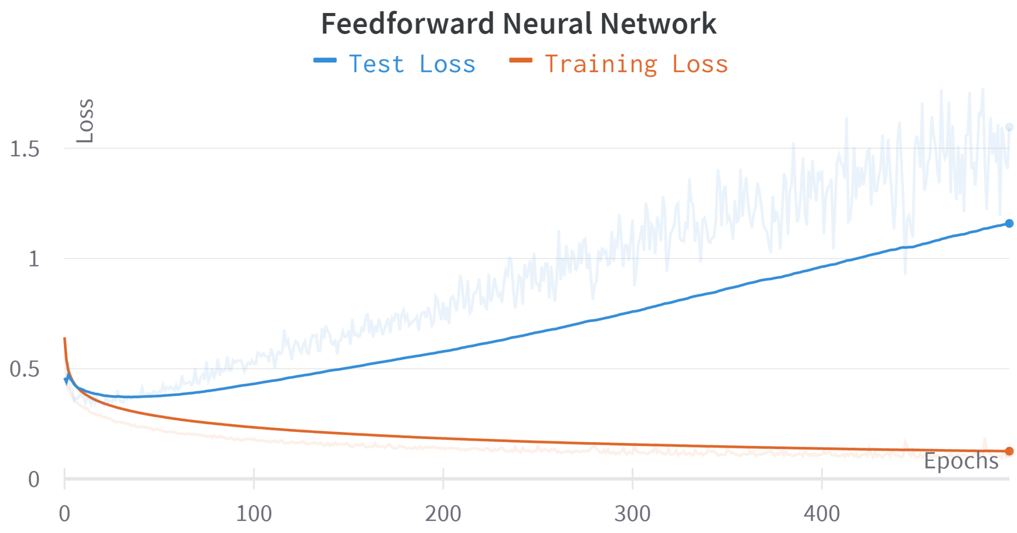


Figure 4.6: FFNN: Train and Test loss during 500 Epochs



We used Dropout with a probability of 0.5 before the final layer of the network so that the model does not overfit in early epochs. The test loss starts decreasing after the 22nd epoch, which indicates that the model has started overfitting the data. We showed the results of the 22nd epoch of FFNN in this paper: see Table 4.1.

class	precision	recall	f1-score
0	0.89	0.92	0.90
1	0.89	0.83	0.83
2	0.86	0.89	0.88

Table 4.1: FeedForward Neural Network: patent classification Precision, Recall, and F1-score

	Class 0	Class 1	Class 2
Class 0	9157	428	386
Class 1	729	8302	1015
Class 2	443	650	8890

Table 4.2: FeedForward Neural Network: Confusion Matrix

FFNN achieves an overall accuracy of 87.83% and an f-1 score of 0.88. The model achieves the best f-1 score of 0.90 for class 0. Confusion Matrix is also shown in Table

4.2 to get better insights into the model's predictions.

We have trained a simple LSTM model for 50 epochs and logged the training and testing accuracy and loss at each epoch. Look at Figure 4.7 and 4.8. We evaluated the model on test-set after each epoch.

Figure 4.7: LSTM: Train and Test accuracy during 50 Epochs

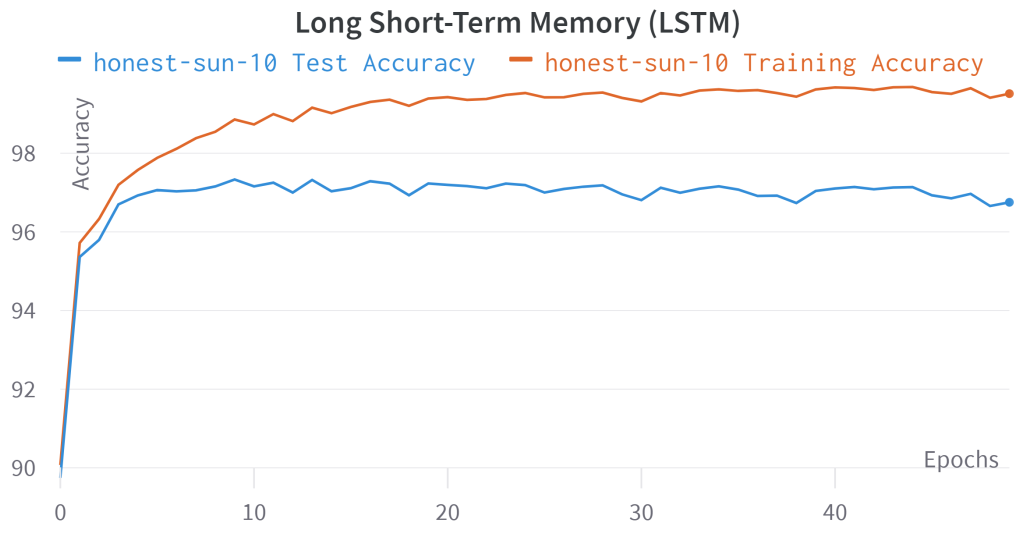
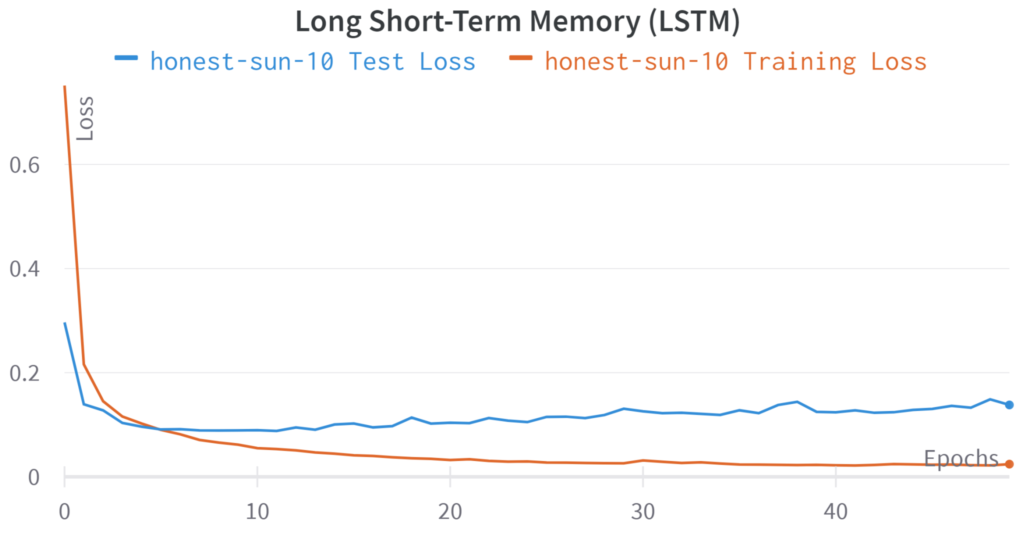


Figure 4.8: LSTM: Train and Test loss during 50 Epochs



It turned out that the 14th epoch gives us the best test accuracy of 97.32% and an f1-score of 0.97. The class-wise precision, recall and f1-score are given in table 4.3 and confusion matrix in Table 4.4. All three classes get an f1-score of 0.97.

We have fine-tuned three pre-trained transformer-based models BERT, RoBERTa, and DistilBERT each for 1 epoch. These are large models so we kept the batch size to 16 to fit it into memory. All three models converge almost at the same rate: see Figure 4.9. The graphs are smoothed by exponential moving averages so that we can easily notice the trend of the loss. After 1 epoch, all three models achieve an f1-score of 0.98. DistilBERT and BERT have an accuracy of 98.17% while the RoBERTa achieves a slightly better accuracy of 98.39%. Considering individual classes,

BERT and DistilBERT achieve an f1-score of 0.98 for all three classes while RoBERTa achieves the best f1-score of 0.99 for class 0. The precision, recall, and f-score for BERT, RoBERTa, and DistilBERT are given in Table 4.7, 4.9, and 4.5 respectively. The confusion matrices are given as Table 4.8, 4.10, and 4.6 respectively.

class	precision	recall	f1-score
0	0.98	0.96	0.97
1	0.96	0.98	0.97
2	0.97	0.97	0.97

Table 4.3: LSTM Network: patent classification Precision, Recall, and F1-score

	Class 0	Class 1	Class 2
Class 0	9608	206	157
Class 1	100	9821	125
Class 2	111	180	9692

Table 4.4: LSTM Network: Confusion Matrix

class	precision	recall	f1-score
0	0.98	0.98	0.98
1	0.98	0.99	0.98
2	0.99	0.98	0.98

Table 4.5: DistilBERT: patent classification Precision, Recall, and F1-score

	Class 0	**Class 1**	**Class 2**
Class 0	9780	114	77
Class 1	56	9921	69
Class 2	103	130	9750

Table 4.6: DistilBERT: Confusion Matrix

class	**precision**	**recall**	**f1-score**
0	0.99	0.98	0.98
1	0.99	0.98	0.98
2	0.97	0.99	0.98

Table 4.7: BERT: patent classification Precision, Recall, and F1-score

	Class 0	**Class 1**	**Class 2**
Class 0	9757	61	153
Class 1	73	9827	146
Class 2	62	55	9866

Table 4.8: BERT: Confusion Matrix

class	**precision**	**recall**	**f1-score**
0	0.98	0.99	0.99
1	0.99	0.98	0.98
2	0.98	0.99	0.98

Table 4.9: RoBERTa: patent classification Precision, Recall, and F1-score

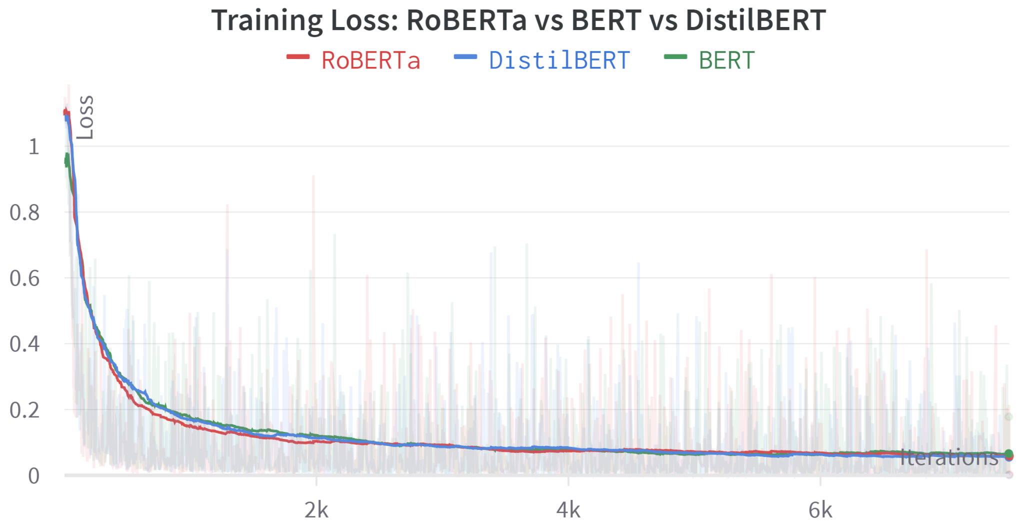
	Class 0	**Class 1**	**Class 2**
Class 0	9829	78	64
Class 1	59	9839	148
Class 2	93	42	9848

Table 4.10: RoBERTa: Confusion Matrix

Model	Precision	Recall	F1-score	Accuracy
FFNN	0.8783	0.8783	0.8783	87.83 %
LSTM	0.9732	0.9732	0.9732	97.32 %
DistilBERT	0.9817	0.9817	0.9817	98.17 %
BERT	0.9817	0.9817	0.9817	98.17 %
RoBERTa	**0.9839**	**0.9839**	**0.9839**	**98.39 %**

Table 4.11: Comparison of our five models

Figure 4.9: Loss: BERT vs RoBERTa vs DistilBERT



We compared the performance of our five different models in Table 4.11. It turned out that RoBERTa performs best in terms of precision, recall, f1-score, and accuracy.

Model	Training time/epoch	# of parameters	Model Size in MBs	# of Epochs	Batch Size	Max Seq Length
FFNN	∼1 second	167,387	0.67 MBs	500	1024	all tokens
LSTM	∼37 seconds	5,761,847	23.04 MBs	40	1024	512 tokens
DistilBERT	1 h 41 m 49 s	65,783,811	263.17 MBs	1	16	512 tokens
BERT	3 h 18 m 55 s	110,075,139	440.385 MBs	1	16	512 tokens
RoBERTa	3 h 26 m 04 s	125,238,531	501.039 MBs	1	16	512 tokens

Table 4.12: Comparison of our five models

We also compared training time per epoch, the total number of trainable parameters, the model size on disk, the total number of training epochs, batch size, and the Maximum sequence length of the models for each model. We have noticed that even though the RoBERTa performs better than all other models in terms of accuracy but it has more number of parameter which increases the training time of the model. Hence one epoch takes 3 hours, 26 minutes, and 4 seconds to train. We also noticed that BERT and DistilBERT have the same performance but DistilBERT is 1.67 times smaller than BERT and has a smaller number of parameters and disk size. DistilBERT trains faster than BERT and RoBERTa.

Model	Precision	Recall	F-1 Score	Accuracy
FFNN	0.88	0.88	0.88	87.83 %
LSTM	0.97	0.97	0.97	97.32 %
DistilBERT	**0.98**	**0.98**	**0.98**	98.17 %
BERT	**0.98**	**0.98**	**0.98**	98.17 %
RoBERTa	**0.98**	**0.98**	**0.98**	**98.39 %**

Table 4.13: Comparison of our five models

Finally, we compared the performance of our models with the base models. LSTM, BERT, RoBERTa, and DistilBERT perform better than all base models by some margin. The Feedforward Neural Network perform better than the RFC model but It can not beat other base models.

Chapter 5

Discussion

This chapter presents a brief discussion of our work, its limitations and future research directions.

5.1 Conclusions

In this work, we performed sentiment analysis and patent detection on the "patent sentiment analysis dataset" using pre-trained multilingual models, i.e., FFNN, LSTM, BERT, DistilBERT and RoBERTa. We have trained different FeedForward Neural Networks (FFNN) with different hyperparameters and logged our precision, recall and f1-score, and confusion matrix. We used pre-trained GloVe embeddings as word embeddings. Sentence embeddings were calculated as the element-wise mean of all word vectors of that sentence.

It turned out that Roberta performs best in terms of precision, recall, f1-score, and accuracy. We also compared training time per epoch, the total number of trainable parameters, the model size on disk, the total number of training epochs, batch size, and the Maximum sequence length of the models for each model. We have noticed that even though the RoBERTa performs better than all other models in terms of accuracy but it has more number of parameter which increases the training time of the model. Hence one epoch takes 3 hours, 26 minutes, and 4 seconds to train.

We also noticed that BERT and DistilBERT have the same performance but DistilBERT is 1.67 times smaller than BERT and has a smaller number of parameters and disk size. DistilBERT trains faster than BERT and RoBERTa. Finally, we compared the performance of our models with the base models. LSTM, BERT, RoBERTa, and DistilBERT perform better than all base models by some margin. The Feedforward Neural Network perform better than the RFC model but it cannot beat other base models. Furthermore, our study showed that using pre-trained models for a limited dataset to improve accuracy is more practical. Large models like DistillBERT and RoBERTa will overfit on little datasets if they are trained from scratch, especially for languages with limited resources.

5.2 Limitations and Future Directions

In this section, we present limitations of our work and possible future directions of further research.

5.2.1 Limitations of Current Study

Based on our study and literature, we discovered that different regions of the world have different patent drafting procedures and formats. For instance, in order to maximize their chances of being granted a patent, applicants from Asian countries tend to explicitly list several benefits of their innovation and disadvantages related to the sources they cited. Patents with such clearly defined structures as specific sections make it easier for examiners and attorneys to swiftly go through documents to find important highlights and debatable subject topics.

5.2.2 Future Work

- For the limitations found in the current study, we started to search for such unique tags that could serve as important entities during the inspection and comparison of any current literature.
- As seen above, three potential tags are chosen, however, they are not common across all patents. In other words, the availability of such tags is frequently dispersed. For instance, for the year 2020, only about 8000 patents were discovered with such explicit text segments out of almost 200000 grants.
- In order to facilitate patent analysis, it is important to generate a dataset having

special text segments along with training models for highlighting the text in any patents automatically lacking particular tags.

Bibliography

[1] IWAYAMA Makoto. “Overview of patent retrieval task at NTCIR-3”. In: *Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering, 2003.* 2003.

[2] Gaetano Cascini and Federico Neri. “Natural Language Processing for patents analysis and classification”. In: *ETRIA World Conference, TRIZ Future.* 2004, pp. 199–212.

[3] Milan Agatonovic et al. “Large-scale, parallel automatic patent annotation”. In: *Proceedings of the 1st ACM workshop on Patent information retrieval.* 2008, pp. 1–8.

[4] Hidetsugu Nanba et al. “Overview of the patent mining task at the ntcir-8 workshop”. In: *NTCIR.* 2008.

[5] Hideo Joho, Leif A Azzopardi, and Wim Vanderbauwhede. “A survey of patent users: an analysis of tasks, behavior, search functionality and system requirements”. In: *Proceedings of the third symposium on Information interaction in context.* 2010, pp. 13–24.

[6] Chao-Chan Wu and Ching-Bang Yao. “Constructing an intelligent patent network analysis method”. In: *Data Science Journal* 11 (2012), pp. 110–125.

[7] Assad Abbas, Limin Zhang, and Samee U Khan. “A literature review on the state-of-the-art in patent analysis”. In: *World Patent Information* 37 (2014), pp. 3–13.

[8] Xu Chen and Na Deng. “A semi-supervised machine learning method for Chinese patent effect annotation”. In: *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.* IEEE. 2015, pp. 243–250.

[9] Antonino Laudani et al. “On training efficiency and computational costs of a feed forward neural network: a review”. In: *Computational intelligence and neuroscience* 2015 (2015), pp. 83–83.

[10] Andrew Rodriguez et al. “Graph kernel based measure for evaluating the influence of patents in a patent citation network”. In: *Expert systems with applications* 42.3 (2015), pp. 1479–1486.

[11] Longhui Zhang, Lei Li, and Tao Li. “Patent mining: a survey”. In: *ACM Sigkdd Explorations Newsletter* 16.2 (2015), pp. 1–19.

[12] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. “Recurrent neural network for text classification with multi-task learning”. In: *arXiv preprint arXiv:1605.05101* (2016).

[13] C. Yang et al. “Requirement-oriented core technological components identification based on sao analysis”. In: *Scientometrics* 112.3 (2017), pp. 1229–1248.

[14] Tabrez Y Ebrahim. “Automation & predictive analytics in patent prosecution: USPTO implications & policy”. In: *Ga. St. UL Rev.* 35 (2018), p. 1185.

[15] Shaobo Li et al. “DeepPatent: patent classification with convolutional neural networks and word embedding”. In: *Scientometrics* 117 (2018), pp. 721–744.

[16] Yingyu Wang et al. “A CNN-based Feature Extraction Scheme for Patent Analysis”. In: *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. 2018, pp. 2387–2391. doi: 10.1109/CompComm.2018.8780690.

[17] Lizhong Xiao, Guangzhong Wang, and Yang Zuo. “Research on Patent Text Classification Based on Word2Vec and LSTM”. In: *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 01. 2018, pp. 71–74. doi: 10.1109/ISCID.2018.00023.

[18] Lea Helmers et al. “Automating the search for a patent’s prior art with a full text similarity search”. In: *PloS one* 14.3 (2019), e0212103.

[19] Jieh-Sheng Lee and Jieh Hsiang. “Patentbert: Patent classification with fine-tuning a pre-trained bert model”. In: *arXiv preprint arXiv:1906.02124* (2019).

[20] Mirjana Pejic-Bach, Jasmina Pivar, and Živko Krstić. "Big data for prediction: patent analysis–patenting big data for prediction analysis". In: *Big Data Governance and Perspectives in Knowledge Management*. IGI Global, 2019, pp. 218–240.

[21] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2019).

[22] Walid Shalaby and Wlodek Zadrozny. "Patent retrieval: a literature review". In: *Knowledge and Information Systems* 61 (2019), pp. 631–660.

[23] Jheng-Long Wu. "Patent Quality Classification System Using the Feature Extractor of Deep Recurrent Neural Network". In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2019, pp. 1–8. DOI: 10.1109/BIGCOMP.2019.8679141.

[24] Jayanta Kumar Dutta et al. *Effective building block design for deep convolutional neural networks using search*. US Patent 10,776,668. Sept. 2020.

[25] Daniel S. Hain et al. "A Text-Embedding-based Approach to Measure Patent-to-Patent Technological Similarity". In: *arXiv preprint arXiv:2003.12303* (2020), pp. 1–45.

[26] Samia Khalid. *BERT Explained: A Complete Guide with Theory and Tutorial.* 2020. URL: https://medium.com/@samia.khalid/bert-explained-a-complete-guide-with-theory-and-tutorial-3ac9ebc8fa7c.

[27] Julian Risch et al. "Patentmatch: a dataset for matching patent claims & prior art". In: *arXiv preprint arXiv:2012.13919* (2020).

[28] Adrian de Wynter and Daniel J Perry. "Optimal subarchitecture extraction for BERT". In: *arXiv preprint arXiv:2010.10499* (2020).

[29] Hamid Bekamiri, Daniel S Hain, and Roman Jurowetzki. "PatentSBERTa: a deep NLP based hybrid model for patent distance and classification using augmented SBERT". In: *arXiv preprint arXiv:2103.11933* (2021).

[30] Rudi NA Bekkers and Arianna Martinelli. "A Network Analysis Approach to Intellectual Property Research". In: *Handbook of Intellectual Property Research: Lenses, Methods, and Perspectives* (2021), pp. 506–522.

[31] Renukswamy Chikkamath et al. “Patent Sentiment Analysis to Highlight Patent Paragraphs”. In: *arXiv preprint arXiv:2111.09741* (2021).

[32] Lintao Fang et al. “Patent2Vec: Multi-view representation learning on patent-graphs for patent classification”. In: *World Wide Web* 24.5 (2021), pp. 1791–1812.

[33] Michael Freunek and André Bodmer. “BERT based freedom to operate patent analysis”. In: *arXiv preprint arXiv:2105.00817* (2021).

[34] Majdi H. Beseiso. “Essay Scoring Tool by Employing RoBERTa Architecture”. In: *International Conference on Data Science, E-learning and Information Systems 2021*. 2021, pp. 54–57.

[35] Ralf Krestel et al. “A survey on deep learning for patent analysis”. In: *World Patent Information* 65 (2021), p. 102035.

[36] Sebastian Kula, Michał Choraś, and Rafał Kozik. “Application of the bert-based architecture in fake news detection”. In: *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020) 12*. Springer. 2021, pp. 239–249.

[37] Effectual Services. *Life Cycle of a Patent*. 2021. URL: https://www.effectualservices.com/life-cycle-of-a-patent/.

[38] Ken Voskuil and Suzan Verberne. “Improving reference mining in patents with BERT”. In: *arXiv preprint arXiv:2101.01039* (2021).

[39] Xindong You et al. “Applying Deep Learning Technologies to Evaluate the Patent Quality with the Collaborative Training”. In: *Wireless Communications and Mobile Computing* 2021 (2021), pp. 1–23.

[40] Hamid Bekamiri, Daniel S Hain, and Roman Jurowetzki. “A Survey on Sentence Embedding Models Performance for Patent Analysis”. In: *arXiv preprint arXiv:2206.02690* (2022).

[41] Roberto Henriques, Adria Ferreira, and Mauro Castelli. “A Use Case of Patent Classification Using Deep Learning with Transfer Learning”. In: *Journal of Data and Information Science* 7.3 (2022), pp. 49–70.

[42] United States Patent and Trademark Office. *Understanding the Patent Examination Process*. 2022. URL: https://www.uspto.gov/sites/default/files/documents/InventionCon2020_Understanding_the_Patent_Examination_Process.pdf.

[43] Dongsheng Zhai et al. “Patent representation learning with a novel design of patent ontology: Case study on PEM patents”. In: *Technological Forecasting and Social Change* 183 (2022), p. 121912.

[44] Author(s). *DistilBERT: Understanding the Intuition, Mathematics, and Applications*. 2023. URL: https://iq.opengenus.org/distilbert/.

[45] Author(s). *Text Classification using CNN*. 2023. URL: https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9.