Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

Master Thesis in ICT for Internet and Multimedia

Applying BERT for Feature Extraction in Battery Management System Domain: A Named Entity Recognition Perspective

Master Candidate

Sevval AZ

Student ID 2041453

Supervisor

Asst. Prof. Federico Chiariotti

University of Padova

Co-supervisor

Alessio Gobbo

Bluewind

Academic Year
2022/2023

I dedicated this thesis to my dear mum for
her endless love, support, and encouragement,
to my dear family,
and to Merve, Gulus, and Esther, my lovely friends
who became my family during this time.

Abstract

Optimizing battery performance, enhancing energy storage systems, and assuring their safe and reliable operation all depend heavily on the analysis of battery data. The widely used Bidirectional Encoder Representations from Transformers (BERT) model is examined in this study with a focus on the Named Entity Recognition (NER) task for feature extraction in battery data. By utilizing BERT, it will be possible to identify and categorize important battery-related assets such safety terminology, events, conditions and digital signal state. A approach is suggested that involves fine-tuning BERT in a battery-specific corpus, taking advantage of the capacity to learn representations from large-scale text data. It is aimed to demonstrate the effectiveness of the methodology in accurately extracting battery-related assets through the application of experiments and evaluations.The study aims to make a significant contribution to accelerating and automating Requirement Engineering (RE) document review by providing a more complex and contextually aware approach to feature extraction through the use of BERT-based architectures in projects developed in the Battery Management Systems (BMS) field.

Contents

List of Figures

List of Tables

List of Acronyms

BMS Battery Management Systems

RE Requirement Engineering

NLP Natural Language Processing

NLP4RE Natural Language Processing for Requirements Engineering

AI Artificial Intelligence

NER Named Entity Recognition

BERT Bidirectional Encoder Representations from Transformers

POS Part-of-Speech

UML Unified Modeling Language

RoBERTa A Robustly Optimized BERT Pretraining Approach

CNN Convolutional Neural Network

ROUGE Recall-Oriented Understudy for Gisting Evaluation

NEs Named Entities

GPT Generative Pre-trained Transformers

BIO Inside outside beginning

LSTM Long Short Term Memory

MLM Masked Language Models

NSP Next Sentence Prediction

API Application Programming Interface

OCR Optical Character Recognition

NLTK Natural Language Toolkit

1 INTRODUCTION

In the contemporary era characterized by rapid societal advancements, it becomes imperative to emphasize the efficacy and reliability of energy storage systems in response to the escalating demand for portable electronic devices, electric automobiles, and sustainable energy alternatives. Battery Management System (BMS) emerges as a pivotal component amidst the other components contributing to the success of these systems. The implementation of BMS is of utmost importance in ensuring the secure and reliable functioning of these systems. Electric cars are playing a crucial part in the current landscape due to their reduced gas emissions and good energy use. In order to ensure optimal performance and the provision of necessary power, the implementation of a robust BMS is required for electric cars, which rely on a multitude of battery cells [12]. In order to ensure driving safety and mitigate the risk of potential failures, it is imperative to conduct a thorough assessment of a battery's overall condition [33]. In addition, the timely identification of inadequate performance holds significant importance for electric cars, as it facilitates rapid repair of the battery system, reduces operational expenses, and mitigates the risks of accidents and malfunctions [17]. Effective battery management is crucial in complex systems, such as those found in the marine sector, as well as in personal applications for everyday usage. The surveillance of health conditions is of utmost importance in ensuring the safety of battery-powered vessels navigating the ocean [29]. Classification societies sometimes require independent evaluations, such as yearly capacity testing. However, an alternative approach is to utilize data-driven state of health modeling, which utilizes operational sensor data from batteries. The aforementioned approach encompasses a comprehensive examination of many data-driven methodologies for estimating the state of health, hence furnishing insights into the current advancements in marine battery systems [29]. Energy management will play a pivotal role in several domains that significantly impact our present and future everyday existence. In this context, BMS will serve as a safeguard, ensuring optimal functionality and dependability of the systems under its purview.

The thorough examination and extraction of information from Requirement Engineering (RE) documents are vital for the development of a system that possesses trustworthy safety functions for BMS. Requirement engineering is often regarded as the most efficacious phase inside the software development process. The process of identifying and extracting requirements, a crucial task in RE, may be accomplished through two main approaches: engaging in discussions with stakeholders or analyzing existing resources such as prior systems or documentation. Another aspect to consider is documentation, which involves the development of requirements and often culminates in the creation of a final requirements document [22]. The RE process acts as a fundamental framework around which the entire project is constructed, facilitating a comprehensive comprehension of the system's intended accomplishments and operational mechanisms. The use of requirements engineering concepts is necessary across every phase of the software development process [20]. When considering critical factors such as the presence of consistency and completeness checks in an existing application, the alignment of required system functions, the inclusion of security functions, adherence to functional security standards, and other relevant considerations, RE plays a vital role in ensuring that the behavior of the system aligns with safety standards and regulatory requirements within the context of BMS safety functions.

Natural Language Processing (NLP) has evolved into a set of extremely efficient technologies for effectively managing the increasing volume of textual data present in RE documents. The combination of NLP with RE, also known as Natural Language Processing for Requirements Engineering (NLP4RE), is a notable approach for effectively handling vast quantities of intricate textual data within the RE domain [35]. The discipline of NLP, which falls under the umbrella of AI [7], encompasses several techniques for facilitating communication between computers and human language. Within NLP-Named Entity Recognition (NE), a specialized branch [24], specifically concentrates on the identification and extraction of specific entities or information from unstructured textual data. In the context of BMS safety, NLP techniques, namely NER, may be employed to extract and systematically categorize safety-related specifications and data from diverse textual resources, including patent filings, scholarly publications, governmental laws, and industry standards.

In this study, a methodology is presented that aimed at extracting relevant features in the BMS domain through the integration NER, one of the NLP techniques, into RE processes. The approach involves a multi-faceted strategy, which includes the development of a custom NER model using BERT, the creation of a specialized dataset through the labeling of patent documents in the BMS domain, and the subsequent utilization of the custom NER model for feature extraction. By combining the power of NLP, custom NER model, and domain-specific dataset, this approach aims to contribute to requirements engineering, one of the most important steps of software development process. This study contributes to the ongoing efforts to ensure the safety and reliability of BMS in the fast-evolving landscape of energy storage systems.

The study unfolds as follows, Chapter 2 will provide an overview of the study's background. A summary of the significance of BMS security functions, their crucial roles in diverse applications, and difficulties with RE for BMS security will be described. The need for NLP-NER techniques to overcome these challenges will be highlighted. Additionally, will be explained how these technologies can be used to enhance the analysis of RE documents in the context of BMS security by digging into the fundamental ideas of NLP and NER. Then, Chapter 3 will describe the procedures used to create a unique NER model using BERT, one of the most sophisticated pre-trained NLP models. It will be described how to tag patent documents to create a domain-specific dataset and how to use this dataset to improve the NER model. The specification extraction and knowledge acquisition phase will be illustrated, showing how the unique NER model is implemented into an example BMS security function requirements analysis. Lastly, in Chapter 4, the study's findings will be reviewed, along with any difficulties that were encountered and any potential effects of the technique. Future directions for NLP4RE's development and use in BMS security services will also be suggested.

2
RESEARCH BACKGROUND

This chapter provides a comprehensive overview of the importance of BMS security functions, their essential roles in various applications, and the challenges associated with requirements engineering for BMS security. The following piece will emphasize the importance of employing NLP-NER strategies in order to address and overcome the aforementioned issues.

2.1 Battery Management Systems

In the preceding chapter 1, the relevance and varied uses of BMS were examined. Considering the significant influence of BMS in many fields and the widely recognized significance of RE in software development, it is crucial to emphasize the essentiality of precisely scrutinizing and comprehending RE documentation in BMS projects. The thorough examination of RE documents is crucial for guaranteeing the favorable result of the project, as it establishes the groundwork for delineating, recording, and thoroughly grasping the functional and non-functional requirements that are vital for the development of BMS. The precise understanding and execution of these specifications not only ensure the intended functionality and performance of the BMS, but also enhance the overall safety, efficiency, and dependability of the systems it serves. Therefore, it is important to conduct a comprehensive analysis and successfully include the RE papers into the project's development procedure in order to attain the project's goals and provide a sturdy, dependable, and efficient BMS.

A multitude of projects in the field of BMS have come to light with each project focusing on the thorough examination of numerical data to enhance battery efficiency. Although the previous projects center around different facets of battery health and operating circumstances, it is important to acknowledge that the integration of NLP into the realm of BMS has not been extensively investigated. The limited number of initiatives focused on the integration of NLP highlights a distinct opportunity to connect the complex field of battery diagnostics with the communicative capabilities of NLP. This connection has the potential to enhance the accessibility and intelligence of BMS.

One of the remarkable and up-to-date projects focusing on NLP integration is BatteryBERT, a pre-trained language model for developing battery databases [15]. The paper presents an examination of BatteryBERT, a collection of pre-trained language models that have been particularly developed to improve battery-related datasets and facilitate text mining in the domain of battery research. The authors emphasize the growing quantity of scientific papers in the field of battery research and the necessity for effective information retrieval and data extraction from these texts. The authors employ BERT models in order to accomplish this objective, as BERT models possess the ability to autonomously interpret scientific material with minimum human intervention. A total of six BERT models were constructed, specifically focusing on battery-related topics. These models include BatteryBERT, BatteryOnlyBERT, and BatterySciBERT. The models underwent training using a corpus consisting of battery research papers. Subsequently, they were fine-tuned to perform specific tasks, including battery paper classification and extractive question-answering for the categorization of battery device components such as anode, cathode, and electrolyte materials. The findings of this study indicate that the BatteryBERT models exhibited superior performance compared to the original BERT models when applied to battery-specific activities. The research further establishes a database augmentation pipeline that integrates transformer-based methodologies with conventional techniques. The transformer-based methodology included the utilization of pre-trained BatteryBERT models for the purpose of classifying documents pertaining to batteries and extracting relevant data, specifically focusing on battery device components. The conventional methodology involved the utilization of a battery property parser for the purpose of extracting data from contemporary literature sources. Subsequently, the two sets of data were merged in order to augment the battery database. The research finishes by highlighting the utility of BatteryBERT models in augmenting battery databases and doing diverse text-mining tasks related to batteries. BatteryBERT is a notable and influential project that centers on the integration of NLP. It is a pre-trained language model specifically designed for the development of battery databases.

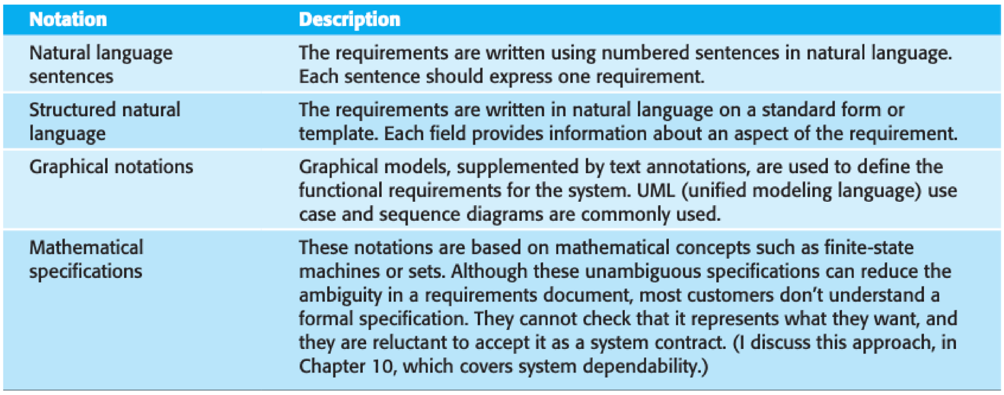
2.2 Requirements Engineering

The definition of the services that a system ought to provide and the constraints on its operational capabilities are explicated in the system's needs. The aforementioned standards are indicative of the requirements expressed by users for a system that is capable of executing various operations, such as placing orders, administering a device, or accessing information. RE is a systematic approach used to discover, analyze, describe, and validate the services and restrictions associated with a particular system or software [25]. In the event that a firm intends to allocate a contract for a substantial software development undertaking, it is imperative for them to articulate their requirements in a manner that is adequately abstract, therefore avoiding the provision of a pre-determined solution. Once a contract has been awarded, it is incumbent upon the contractor to furnish the client with a more comprehensive system specification, therefore enabling the customer to validate the functioning of the program. These two documents may alternatively be denoted as the system requirements document. The distinction between user needs and system requirements lies in the terminology used to refer to each. User requirements pertain to high-level abstract requirements, whereas system requirements encompass a detailed description of the desired functionality of the system.

The following are examples of user needs and system requirements:

- User requirements refer to the boundaries of the services that a system is expected to provide to its users, as well as the constraints that must be adhered to. These requirements are often articulated in normal language and may be supplemented by visual aids such as diagrams and tables. The user requirements might vary from high-level statements outlining the essential system features to detailed and comprehensive descriptions of the system's operation.
- System requirements encompass comprehensive descriptions of the functionalities, services, and operational constraints inherent in the software system. The system requirements document, sometimes referred to as a functional specification, is intended to provide a comprehensive and detailed description of the implementation plan. The specification about

this matter may be included inside the contractual agreement established between the software developers and the buyer of the system. In addition to being represented through various means such as forms, images, or mathematical system models, system requirements may also be communicated in natural language, similar to user needs. The various formats for documenting writing system requirements are illustrated in Fig.2.1.



Notation	Description
Natural language sentences	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system. UML (unified modeling language) use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want, and they are reluctant to accept it as a system contract. (I discuss this approach, in Chapter 10, which covers system dependability.)

Figure 2.1: Notes for writing system requirements [25]

Functional and non-functional requirements are two common categories for software system requirements:

- Functional requirements contain explicit statements on the system's requisite functionalities, delineating its expected responses to certain inputs and actions to be taken under specified circumstances. In certain cases, the functional requirements may also include explicit descriptions of what tasks the system should refrain from accomplishing.
- Non-functional requirements contain the constraints placed upon the system's capacity to deliver certain services or perform tasks. The limits of these systems encompass standards-based constraints, restrictions on the development process, and temporal limitations. Non-functional requirements sometimes pertain to the entirety of the system, rather than focusing on individual features or services inside the system.

Contrary to what these straightforward definitions imply, the distinction between unalike types of requirements is not always clear-cut. A system's functional requirements statement should ideally be comprehensive and consistent. Completeness refers to the definition of all services and data that the user may need. Requirements should not conflict with one another in order to be consistent [36]. In reality, very tiny software systems are the only ones for which requirements consistency and completeness can be achieved. One reason is that developing specifications for big, sophisticated systems may be difficult and rife with errors and omissions. Non-functional requirements, as the name implies, are specifications that have little to do with the particular services the system provides to its users. These non-functional requirements usually specify or constrain characteristics of the system as a whole. They might be connected to characteristics of emergent systems as dependability, speed of reaction, and memory usage. Alternatively, they could specify limitations on how the system will be implemented, like the capabilities of I/O devices or the data representations used in system interfaces. In many cases, non-functional needs are more important than specific functional requirements. Metrics for defining non-functional needs include speed, size, ease of use, robustness, reliability, and portability [21] . These properties can be measured while testing the system to check whether the system meets it is non-functional properties. Users of the system can frequently find workarounds for system features that do not exactly fit their needs. However, if a non-functional criterion is not met, the entire system may become inoperable. For instance, if an aircraft system does not pass reliability tests, it will not be approved as safe to fly; if an embedded control system does not pass performance tests, the control functions will not work as intended [6]. For all these reasons, it is crucial to correctly extract the functional or non-functional requirements of a system.

To summarize, the initial stage of the software engineering process is commonly represented as RE. Prior to proceeding with the acquisition or development of a system, it is imperative to get a comprehensive comprehension of the system's needs. This preliminary RE provides a comprehensive overview of the potential achievements and benefits that the system might potentially deliver. These factors can be included in a feasibility study, which seeks to assess the technical and financial viability of the system. The results of the aforementioned study provide valuable insights for management in making informed decisions about the continuation of system development or the acquisition of a new system. Due to its significance, RE has tremendous relevance throughout the beginning phase of any project, including those within the BMS area.

2.3 NLP Studies in RE Documents

Since the 1950s, software requirements have commonly been expressed using natural language [25]. The phenomenon under consideration exhibits a wide prevalence, intuitive nature, and expressive qualities. Moreover, the level of clarity and potential for confusion in the text may vary depending on the reader's background, leading to multiple possible interpretations. Consequently, a multitude of alternative methodologies for composing requirements have been proposed. Given the lack of widespread adoption of any alternative approaches, it may be inferred that natural language will persist as the predominant means of articulating system and software needs.

One notable investigation exploring the application of NLP in the domain of RE is the comprehensive mapping study undertaken by Zhao et al. [35].This mapping study provides a thorough and extensive examination of the research conducted in the subject of NLP4RE. It gives significant insights that can inform and guide future work in this area. The research effort involved reviewing 404 main papers from a pool of 11,540 search results, with the aim of addressing five fundamental research inquiries. The field of NLP4RE has experienced substantial growth and garnered considerable interest from diverse sectors, as evidenced by its extensive publication record and widespread recognition. Notwithstanding several constraints, the study also emphasizes the advancements achieved in the field of NLP4RE research throughout the preceding 15 year period, particularly with regard to scholarly publications and the creation of tools. The current endeavors to examine complex documents and the growing interest from industries in advanced NLP technologies suggest that research on NLP4RE has the potential to evolve into a viable tool for facilitating requirements engineering practices. One of the primary inquiry of this research is to the NLP technologies that allow NLP4RE research. The Fig.2.2 illustrates the utilization of the most commonly used approaches in NLP, as concluded from the given query.

According to the data presented in Fig.2.2, it can be observed that the NLP approach that has been employed most frequently is Part-of-Speech (POS) Tagging, as indicated by 187 times. Following POS Tagging, Tokenization has been utilized 81 times, Parsing 72 times, Stop-Words Removal 70 times, Term Extraction 68 times, and Stemming 68 times. According to the findings presented in

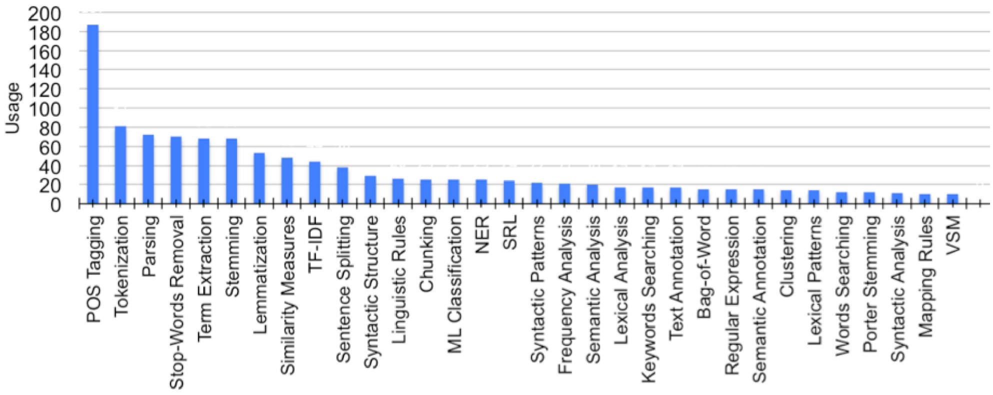


Figure 2.2: Techniques in NLP that have been utilized at least 10 times [35]

Fig.2.2, it can be observed that a majority of the approaches employed in NLP, particularly those that are widely utilized, are primarily focused on syntax. This observation serves as a significant indication of the predominant role played by syntactic techniques in the field of NLP4RE research.

The data shown in experiment highlights the importance of integrating syntax-related elements into the process of NER. The accuracy of entity identification might possibly be improved by including techniques such as POS Tagging and Parsing, which allow for the capture of syntactic context and connections pertaining to entities. Moreover, the widespread usage of Tokenization, Stop-Words Removal, and term extraction implies that these preprocessing techniques play a vital role in the preparation of text data for NER tasks. Therefore, it is essential to incorporate these procedures into feature extraction pipeline. The interconnection of NER, POS Tagging, and Tokenization lies in their sequential integration inside the NLP pipeline. Tokenization is commonly regarded as the initial stage in which text is divided into smaller units known as tokens. POS tagging is a process that involves assigning grammatical labels to each token in order to provide information about its specific grammatical role. In conclusion, the process of NER utilizes the tokenized and POS-tagged text in order to detect and categorize named entities. These activities are frequently executed consecutively in order to get organized data from text that lacks a predetermined framework.

In a nutshell, the results depicted in Fig.2.2 provide valuable guidance for the methodology employed in our study. They serve as a source of inspiration for us to investigate and integrate these approaches into our feature extraction methodology, with the objective of enhancing the performance of NER and attaining a more thorough comprehension of entities in text written in natural language.

An additional exemplification of a study in the field of NLP within the framework of RE is shown in the article produced by Alkhader et al. [1]. This study presents a novel framework for the automation of software requirements extraction through the utilization of NLP techniques. The primary objective of their framework is to accept English natural language requirements as input and autonomously produce appropriate Unified Modeling Language (UML) class diagrams utilized for the representation of software architectures. Furthermore, the primary objective of the framework is to enhance the feasibility of reusing specifications by means of the reverse engineering procedure, hence optimizing the efficiency and reducing the workload for requirements engineers. The significance of this study is within the domain of software engineering, more especially in the realm of RE. The phase of RE has significant importance in the process of software development, as it encompasses the tasks of gathering, assessing, and documenting system specifications along with functional and non-functional needs. The utilization of NLP inside this particular scenario has several significant benefits.

The benefits can be listed as follows:

1. Automation: The use of NLP can effectively mitigate the need for human labor by automating the process of extracting and analyzing requirements from documents written in natural language.

2. Efficiency is a notable advantage since it greatly expedites the conversion of textual requirements into structured, machine-readable representations, such as UML class diagrams.

3. Accuracy: The application of NLP approaches has the potential to enhance the accuracy of requirements extraction by mitigating the potential for misunderstandings or misinterpretations among stakeholders.

4. Consistency: The utilization of NLP might provide a higher level of consistency and standardization in the domain of RE, hence promoting uniformity across various projects.

5. Reusability:The utilization of NLP can facilitate the identification of patterns and similarities in requirements, hence enhancing the potential for

feature reuse across diverse projects and ultimately augmenting the efficiency of software development as a whole.

6. Adaptability:The adaptability of NLP frameworks enables their utilization across diverse natural languages, rendering them well-suited for international software development endeavors.

In conclusion, the utilization of NLP within the context of RE documentation has significance due to its ability to speed the process, enhance precision, and facilitate the extraction of valuable insights from text composed in natural language. This phenomenon can lead to enhanced efficacy in software development processes and improved intercommunication among developers and stakeholders.

Yet another study, Herwanto et al. [14], underlines the significance of employing NLP methodologies within the domain of RE, specifically emphasizing the aspect of privacy needs. The issues related to incorporating privacy concerns into the software development process are emphasized by the authors, especially in the context of strict rules like the General Data Protection Regulation. The article presents a new automated technique that utilizes NLP, especially NER, to detect privacy-related elements in user tales. The entities in question consist of the Data Subject, Processing, and Personal Data. The major aim of the automation is to improve the modeling process, hence improving the efficiency of RE. The NER model and embeddings are key components in NLP tasks. The authors utilize state-of-the-art NER models in conjunction with contextual embedding approaches to detect entities linked to privacy. The authors give empirical results that compare different embedding techniques and showcase the enhanced efficacy of transformer models, such as BERT and A Robustly Optimized BERT Pretraining Approach (RoBERTa), within this particular domain. The NER model is employed to analyze user tales derived from the Solid Project, demonstrating its effectiveness in identifying Data Subjects, Processing components, and Personal Data entities within these narratives. In conclusion, the research emphasizes the significant importance of NLP approaches, particularly named entity recognition NER, in the automation of identifying privacy-related entities within user narratives. The implementation of automation in privacy RE has been shown to improve both the efficiency and efficacy of the process, especially in agile software development contexts.

In another article [16], the classification problem, one of the frequently used methods of NLP, is focused on the field of RE. Requirements classification, which is a crucial task in RE, is challenging because requirements are often expressed in natural language. Non-functional requirements are quality attributes that play a critical role in software success. The paper presents a model for requirements classification using a BERT-Convolutional Neural Network (CNN), where BERT is utilized to extract features from requirements and CNN extracts depth features. The authors tested the BERT-CNN model on a dataset containing 625 requirements and discovered that it performed better than state-of-the-art methods. It emphasizes the BERT model's use of transfer learning to improve NLP performance. In order to improve needs classification, the BERT-CNN model combines the best features of BERT and CNN. In conclusion, the study highlights the role of NLP in RE, and provides a BERT-CNN model as a viable strategy for automated requirements classification.

Similar to the preceding article [16], another exemplar article [34] provides an examination of non-functional requirements utilizing BERT and addressing the classification problem. A methodology has been developed through which BERT and Topic Model are used to extract non-functional requirements and analyze user reviews, with the aim of enhancing software quality. User reviews of mobile applications offer developers with useful input, including insights into non-functional criteria such as usability, reliability, performance, and supportability. A comprehensive grasp of these needs is crucial for enhancing software. The present article provides an overview of the current methodologies employed in the study of user reviews, which commonly include classification techniques and sentiment analysis. The primary objective is to categorize user reviews into distinct classifications, including problem reports, feature requests, and user experience evaluations. The majority of prior research employs multi-class classification techniques to analyze user reviews. However, this study introduces a novel strategy by utilizing multi-label classification, enabling the categorization of a review into many Non-Functional Requirement categories continuously. The utilization of this technique is crucial as it allows for the simultaneous consideration of several non-functional requirements inside a single assessment. This study presents the use of the BERT model in the context of multi-label categorization of user reviews. BERT, a cutting-edge NLP technology, demonstrates remarkable advancements in accurately recognizing non-functional requirements inside user evaluations. Following the process of categorization, utilizes Latent Dirichlet Allocation to extract themes from the reviews and subsequently do further research. This enables developers to gain a comprehensive understanding of individual concerns highlighted in reviews and to discern customer requirements in a more detailed and precise manner. The primary advancements offered by the suggested approach encompass enhanced multi-label categorization via BERT and a more comprehensive comprehension of customer requests through topic analysis. This methodology facilitates the expeditious comprehension of user requirements by developers, hence resulting in enhanced efficacy in software development and maintenance. This methodology has the potential to optimize developers' time and resources in acquiring and comprehending user feedback, ultimately resulting in enhanced software development and maintenance outcomes.

Another study, which is one of the main reference articles of this study, is the NER study developed using BERT for Aerospace RE, called aeroBERT-NER [27]. This article presents an in-depth review of a project that focuses on the development of a custom NER model called aeroBERT-NER. The primary objective of this research is to create a customized NER model that is specifically tailored for aerospace-related terminology. This is achieved through the process of fine-tuning the model using an annotated corpus that consists of aerospace-specific data. The model exhibits a higher level of performance in comparison to a general NER model, namely BERTBASE-NER, when employed in the analysis of aerospace texts. The primary objective of this study is to construct a model capable of discerning Named Entities (NEs) within aviation criteria. The NEs, contain many categories such as System (SYS), Value (VAL), Date time (DATETIME), Organization (ORG), and Resource (RES). The initial stage involves the development of an annotated aerospace corpus. Publicly accessible aerospace books, such as publications from the National Academy of Aerospace Studies Board and Title 14 of the Code of Federal Regulations, are utilized due to the private characteristics of aerospace requirements. Once the requisite preprocessing procedures have been implemented, the aeronautical corpus undergoes annotation using the BIO tagging method. Subsequently, the corpus is partitioned into distinct NEs classes, namely SYS, VAL, DATETIME, ORG, and RES, each assigned with appropriate labels. The provided annotated aircraft corpus is utilized as input for the purpose of fine-tuning the BERT language model. The fine-tuning procedure involves utilizing datasets of varying sizes, spanning from 250 to 1423 sentences. The F1 scores are employed to assess the performance of the model across different sizes of datasets, hence identifying patterns in the learning process. The F1 score is a metric commonly used in binary and multiclass classification tasks to evaluate the performance of a model. The model known as aeroBERT-NER has a weighted average F1 score of 0.92. The demonstration of aeroBERT-NER's higher performance in the aviation text is achieved by a comparison with bert-base-NER. The research showcases the capacity to convert natural language requirements into forms that can be interpreted by machines, hence facilitating the utilization of model-based methodologies in the field of aeronautical engineering. This paper outlines a complete methodology for constructing aeroBERT-NER, a specialized NER model tailored for aerospace-related terminology. The model is fine-tuned using a meticulously annotated corpus particular to the aerospace domain. The model exhibits a higher level of performance in comparison to a general NER model, namely bert-base-NER, when employed in the analysis of aerospace texts. The research demonstrates that it is possible to develop a similar customized model through the process of fine-tuning BERT models using a dataset that is generated by annotating the domain-specific corpus along with its NER tags.

In summary, the integration of NLP into the domain of RE documentation has significant potential to optimize and improve the RE analysis phase, one of the important steps of the software development life cycle. Numerous significant studies have provided evidence of the utilization of NLP techniques for the purpose of automating the extraction and analysis of requirements from texts written in natural language. Additionally, these techniques have been employed to enhance the classification of requirements and streamline the identification of domain-specific entities within specialized domains such as aerospace engineering. The utilization of NLP in the field of RE has several advantages including automation, enhanced efficiency, improved accuracy, consistent outcomes, reusable artifacts, and adaptable solutions. The ongoing development of NLP4RE highlights the significant contribution of NLP technologies in enhancing the efficiency, accuracy, and overall efficacy of requirements engineering practices. This, in turn, facilitates improved software development processes and encourages enhanced communication between developers and stakeholders.

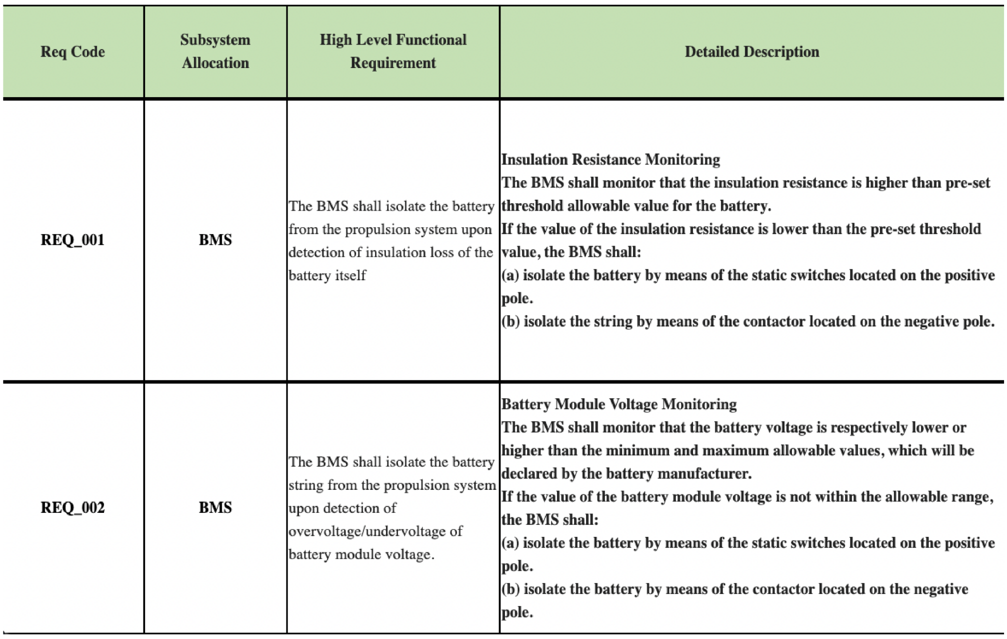
2.4 Patent Documents and RE Documents Similarity

Patent documents and RE documents, although having distinct functions within the domains of intellectual property and software development respectively, exhibit noteworthy parallels in terms of linguistic structure, utilization of claims, and arrangement of material. The presence of these shared characteristics renders patent papers a significant asset for the extraction of features through the utilization of NER in requirement documents intended for training sets. Both patent filings and RE documents utilize an organized and detailed language. The prevalence of this phenomenon arises from the necessity to effectively communicate intricate notions and principles in a lucid and unequivocal fashion. In these fields, it is imperative to employ technical language, specific vocabulary, and well-defined grammar. Patent documents exhibit a noticable standardization in their linguistic structure, characterized by the inclusion of certain parts such as "Background of the Invention," "Summary," and "Claims." The aforementioned parts serve to delineate the extent of the innovation and offer a succinct portrayal of its distinctive attributes. In a similar vein, papers pertaining to requirements engineering adhere to a predetermined framework, frequently encompassing parts such as "Introduction," "Requirements Specification," and "Use Cases" in order to methodically gather and communicate needs.

The key component of patent documents is constituted by the claims, which serve to delineate the unique privileges given to the inventor. The aforementioned claims consist of carefully formulated statements that accurately outline the unique characteristics and capabilities of the invention. The utilization of the phrase "claims" may not be prevalent in RE documents; nonetheless, these documents do encompass assertions that delineate the characteristics, functions, and limitations of a certain system or program. The function of both claims in patents and statements in requirements documents is to delineate the extent and particulars of what is being safeguarded (in the context of patents) or devised (in the context of requirements).

Patent documents adhere to a specified organizational framework, encompassing several components such as the abstract, background, summary, comprehensive description, and claims. In a similar vein, documents pertaining to RE are conventionally structured to encompass an introductory section, a system overview, user requirements, functional requirements, non-functional requirements, and limitations. The establishment of a parallel structure in these two document forms guarantees the systematic and coherent presentation of information, a critical factor in facilitating successful communication and comprehension. The aforementioned commonalities provide patent papers a significant resource for the purpose of training NER models in the field of RE. Through the utilization of the linguistic structure, utilization of claims, and categorization of material present in patent papers, it is possible to extract significant aspects and entities that hold relevance to the advancement of software systems. Patent documents possess the potential to provide a substantial reservoir of technical lexicon, terminology, and organized data. Applying this approach has the potential to improve the performance of NER models when utilized in the context of RE. For example, during the process of collecting software requirements, NER models have demonstrated enhanced proficiency in identifying technical terminology, product nomenclature, and particular functionality by using the linguistic patterns included in patents.

To elucidate the close connection between RE documents and patent content, proceed to examine an exemplification within the domain of BMS. The Fig.2.3 displays examples of requirement items included in a RE document pertaining to BMS domain.

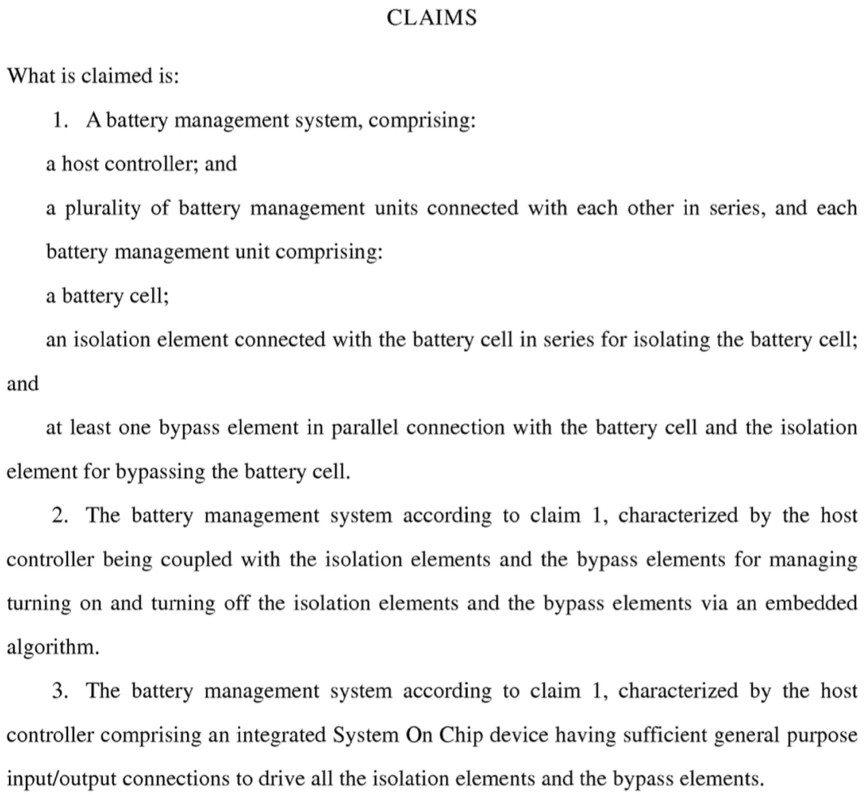


Req Code	Subsystem Allocation	High Level Functional Requirement	Detailed Description
REQ_001	BMS	The BMS shall isolate the battery from the propulsion system upon detection of insulation loss of the battery itself	Insulation Resistance Monitoring The BMS shall monitor that the insulation resistance is higher than pre-set threshold allowable value for the battery. If the value of the insulation resistance is lower than the pre-set threshold value, the BMS shall: (a) isolate the battery by means of the static switches located on the positive pole. (b) isolate the string by means of the contactor located on the negative pole.
REQ_002	BMS	The BMS shall isolate the battery string from the propulsion system upon detection of overvoltage/undervoltage of battery module voltage.	Battery Module Voltage Monitoring The BMS shall monitor that the battery voltage is respectively lower or higher than the minimum and maximum allowable values, which will be declared by the battery manufacturer. If the value of the battery module voltage is not within the allowable range, the BMS shall: (a) isolate the battery by means of the static switches located on the positive pole. (b) isolate the battery by means of the contactor located on the negative pole.

Figure 2.3: A sample RE document written in natural language in the BMS domain

Within the domain of BMS, the reliance on natural language is exemplified by the presence of first and second high level functional requirements in the RE document. An example of a high-level functional requirement is the specification that the BMS should initiate isolation of the battery upon detection of insulation loss. This requirement is further detailed in first description, which provides additional information on the criteria for monitoring insulation resistance and outlines the corresponding actions to be executed. In a similar vein, second high level functional requirement outlines the prescribed behavior of the BMS in relation to instances of overvoltage or undervoltage in the voltage of the battery module, as expounded upon in second detailed description. These requirements serve as a prime example of the significance of natural language within the domain of RE.

Additionally, instance of patent claims in the BMS field can be presented as in the Fig.2.4.



CLAIMS

What is claimed is:

1. A battery management system, comprising:

a host controller; and

a plurality of battery management units connected with each other in series, and each battery management unit comprising:

a battery cell;

an isolation element connected with the battery cell in series for isolating the battery cell; and

at least one bypass element in parallel connection with the battery cell and the isolation element for bypassing the battery cell.

2. The battery management system according to claim 1, characterized by the host controller being coupled with the isolation elements and the bypass elements for managing turning on and turning off the isolation elements and the bypass elements via an embedded algorithm.

3. The battery management system according to claim 1, characterized by the host controller comprising an integrated System On Chip device having sufficient general purpose input/output connections to drive all the isolation elements and the bypass elements.

Figure 2.4: A sample section from the claims of the BMS and Method patent [1]

The notable resemblances seen in the structure of language, utilization of claims, and arrangement of material between patent documents and RE documents emphasize the possibility of using patent documents to augment NER in the field of RE. By utilizing patent material for the purpose of feature extraction and training of NER models, software development teams have the potential to enhance the precision and effectiveness of extracting crucial entities and information from papers containing requirements. The aforementioned cross-domain synergy serves as a prime example of the wide-ranging applications of language analysis, hence showcasing the significance of interdisciplinary methodologies in the fields of information retrieval and knowledge extraction.

The below studies can be cited as exemplars of feature extraction research performed using patents.

The developed study examines the application of NLP in the domain of patent texts for the purpose of extracting inventive information that may be utilized for supporting manufacturing model cases [5]. Patents are often regarded as a valuable repository of innovative knowledge. The authors have integrated the Theory of creative Problem Solving (TRIZ) NLP in order to automate the process of extracting creative information from patent documents. TRIZ is an innovative methodology developed with the aim of enhancing the efficacy of the inventive process. The study examines the creation of an Application Programming Interface (API) that use a hybrid approach, integrating morphosyntactic analysis and machine learning, to extract innovative details from patent documents. The implementation of automation expedites the process of problem-solving. The authors of the study conducted a comparative analysis between their automated extraction method and a manual information collecting strategy lead by humans. The findings indicate a significant 36% improvement in the extraction of valuable information, while also reducing the amount of time required by experts. The paper provides a comprehensive overview of the methods employed in the automatic extraction of creative information from patent documents. This includes a detailed description of the many steps involved, such as text preparation, tokenization, lemmatization, and the elimination of stop words. In the present case study, the authors successfully extracted a total of 142 valid parameters. These parameters serve the purpose of identifying potential conflicts within the patent language. The authors of the study conducted a comparison between the results obtained from an automated extraction method and a solely human-driven approach. The findings of the study indicate that the automated extraction method exhibits more efficiency in comparison to the human-driven strategy. The study's findings suggest that the proposed methodology streamlines the process of problem-solving by automating the extraction of essential components required for populating the ontology. This not only saves time but also leads to a more thorough inclusion of material. Nevertheless, the authors recognize the necessity of enhancing recollection and automating the discovery of contradictions in order to facilitate more methodical innovation processes. In summary, the article focuses on how NLP can be used to automatically extract information from patent contents, particularly in the context of TRIZ and IDM methodologies, and highlights the benefits and challenges of this approach.

An strengthened CNN is used in a new method for patent analysis, which is discussed in an article that extracts features from patent documents and gets remarkable outcomes [31]. It introduces an innovative approach to patent analysis, utilizing an enhanced CNN for extracting features from patent text. The conventional method of patent analysis has predominantly employed probability analysis. However, this suggested methodology incorporates text mining algorithms and Vector Space Modeling to detect abstract "topics" within a set of patent papers. The suggested methodology integrates statistical modeling and CNNs in order to extract sophisticated characteristics from patent text. The CNN algorithm is employed to analyze structured patent data, resulting in the generation of high-dimensional vectors. These vectors offer a greater level of detail and sophistication in comparison to traditional human approaches. The CNN model is commonly employed for the purpose of feature extraction. The process incorporates techniques such as word embedding, convolution, pooling, and full-connection layers to effectively extract intricate information from the textual data. The article describes the findings derived from a series of experiments done using a dataset comprising patents related to telecommunication technology. CNN model demonstrated a notable level of accuracy, reaching 92% in both the training and test datasets. The collected feature vectors were presented in a two-dimensional space, providing a demonstration of the efficacy of the technique. The proposed methodology presents several benefits in comparison to conventional methodologies, as it produces dimension vectors that can be utilized for quantitative analysis. Additionally, it mitigates the occurrence of mistakes in manual data processing and enables the detection of new technologies inside patent documents.

In addition to these studies, multi-label classification, graph structure-based patent mining and patent summarization studies can be briefly given as three examples of NLP studies carried out on patent documents. The primary aims of this study [9] are to create an interactive visualization tool for presenting information contained inside patent documents and to design a highly accurate, multi-label classification algorithm capable of assigning patents to numerous collaborative patent classification categories. The case study encompassed both metadata and text data pertaining to a total of 17,500 patents related to electric vehicles. In order to achieve the desired objectives, the present study employs the following methodology: it relies on the process of feature engineering to extract subjects from patent documents. Furthermore, the data was utilized to train multi-label versions of traditional machine learning algorithms, with the objective of predicting various class labels associated with a certain electric car patent. The findings of this study exhibited encouraging outcomes, attaining high ratings in relation to performance metrics. The results yield a dual outcome. Firstly, it illustrates the efficacy of utilizing open-source NLP technologies in constructing tailored patent analysis pipelines. Additionally, this research contributes to the progress of automating the categorization of patents into Cooperative Patent Classification classes, specifically focusing on multi-label classification. The utilization of this automated classification approach not only optimizes the study of patents but also improves the effectiveness of classifying patents into appropriate Cooperative Patent Classification classes.

The second research [13] use NLP and graph data modeling approaches to derive functional representations of patents pertaining to mechanical designs. The suggested methodology presents a number of significant benefits: The utilization of schema-free graph modeling enables the representation of patent data in a manner that is characterized by flexibility and adaptability. The optimization of data storage and the improvement of query performance for data relationships, the incorporation of visualization libraries enables seamless utilization, hence obviating the requirement for intricate data extraction software. The examination of patents and design concepts facilitates the identification and analysis of commonalities and intersections between the two. This work showcases the efficacy of NLP and graph data modeling in constructing a semantic library of patents, hence enhancing comprehension of mechanical design advancements. The method of manually abstracting patents is frequently characterized by its time-consuming nature and subjectivity, rendering it progressively less effective as the scope of the patent knowledge area expands.

The third research [28] utilizes NLP, deep learning methodologies, and machine learning algorithms to extract crucial information from patent papers within a specified area. This study presents a novel approach to patent summary that incorporates intelligent techniques, allowing for efficient and unbiased summarization of patents, even in fields characterized by vast quantities of patent documents. The system utilizes machine learning techniques to autonomously produce summaries of patent documents. It places particular emphasis on identifying and including essential technical terms within the semantic framework of the training patents and their associated summaries. The study used one of the evaluation metrics of NLP studies called Recall-Oriented Understudy for Gisting Evaluation (ROUGE) measures to assess the precision, recall, accuracy, and consistency of the information produced by the summarization method. Case studies conducted in the field of smart equipment technologies have exhibited remarkable outcomes.

The transformational power of NLP and machine learning in patent analysis is demonstrated by these three NLP experiments on patent documents. They bring attention to the ability to extract, categorize, and summarize patent information, which may be used to gain new insights and boost innovation. The use of NLP is making patents easier to navigate, which in turn accelerates technological and industrial progress.

2.5 NLP and NER Fundamentals

This section presents a detailed overview of NLP and NER. This discussion will address the significant milestones in the domain of NLP, as well as go into the commonly acknowledged practical phases used for NLP. Next, move on to NER, a critical aspect of NLP that involves identifying and classifying entities such as names, locations, and organizations within a given text. This study aims to investigate the concept of transfer learning for NER, which involves utilizing pre-trained models to enhance the accuracy and effectiveness of NER tasks. The BERT model is presented. The use of BERT will be elucidated, as it has significantly transformed the comprehension of language by including bidirectional contextual word analysis, resulting in notable progress in diverse NLP applications.

2.5.1 Evolution of NLP

As stated earlier chapter in 1, NLP, a subfield of Artificial Intelligence (AI), deals with analyzing of diverse linguistic phenomena in order to enhance communication between computers and human language through the application of machine learning methods. It can be asserted that NLP has been employed to address a wide range of tasks, notably including Sentiment Analysis, Text Classification, NER, Text Summarization, and Topic Modeling. Additionally, NLP has been applied to Machine Translation, Text Generation, Speech Recognition, Question Answering, Language Modeling, and Chatbots, which have obtained significant attention in recent years. There are many reasons, especially developing technology, behind the remarkable increase in the use of NLP, which dates back to the 1960s [32], in recent years. The progress in hardware, like as GPUs and TPUs, has sped up the training process of deep learning models, facilitating the development of larger and more robust NLP models. Additionally, this has resulted in a greater accessibility of large textual datasets for the purpose of training models. The utilization of large datasets has facilitated the acquisition of linguistic patterns by models in a highly efficient manner. As can be seen in Fig. 2.5, the area of NLP has seen significant evolution and innovation in the past two decades, driven by these mentioned breakthroughs. The beginnings of NLP in 2003, which signified the shift from rule-based systems to data-driven methodologies, can be seen as the initial stage in this progression [4]. This evolution continued in 2008 with the publication of multitask learning, which included training models to perform multiple NLP tasks simultaneously, such as part-of-speech tagging and named entity recognition [8]. The development of word embeddings in 2013 was a significant milestone in the field of NLP [19] [18]. Word2Vec, a breakthrough innovation, was created by Tomas Mikolov and his colleagues at Google. The semantic links between words are represented by word embeddings, which represent words as dense vectors in a continuous vector space. This major development brought about a significant transformation in the area by enabling models to enhance their comprehension of context and the similarities between words. Subsequently, in the year 2014, deep learning emerged as an established approach in the field of NLP [26]. The progression of models extended beyond basic designs and started the integration of neural networks with several layers. A major turning point in the field of NLP occurred in 2014 with the development of the attention mechanism [3]. The method, which gained popularity through the publication titled "All You Need is Attention" [30] aimed to tackle the issue of managing long-term dependencies within arrays. Attention models, such as the Transformer architecture, offer a more efficient approach for capturing contextual information and word associations, resulting in enhanced performance across many NLP applications. And lastly, the year 2018 was a significant milestone in the progression of NLP. The use of transformer-based models such as industry leaders Google's BERT [11] and OpenAI's Generative Pre-trained Transformers (GPT) [23] is truly a breakthrough in the field of NLP, leaded to rapid development. These models have achieved notable progress by means of learning in diverse manners. Transformer models have the capability to effectively process enormous amounts of text, enabling them to capture the nuances and complexities of language. Pre-training facilitates the transferability of models, allowing for fine-tuning on specific NLP tasks. This methodology enables an important portion of the cost to be allocated towards comprehensive labeling for the specific task, hence enhancing the efficiency of NLP development. Transformers possess a notable degree of scalability, enabling their size to be increased by incorporating additional parameters and layers. Models such as GPT exhibit a substantial scale, characterized by an extensive parameter count reaching into the hundreds of billions. The capacity to scale enables them to effectively catch nuanced linguistic patterns and achieve better results across a diverse array of tasks. Moreover, BERT and GPT demonstrate versatility and have been utilized effectively in a diverse array of NLP tasks. These systems have the capability to execute various tasks, including but not limited to text changes, sentiment analysis, language translation, summarization, question answering, and chatbot functionality. The aforementioned versatility renders them highly advantageous in a multitude of industries, extending from the healthcare sector to the world of e-commerce. Both Google and OpenAI have provided access to their models via APIs, enabling developers to simply include these powerful NLP applications. This accessibility opens up access to state-of-the-art NLP tools. Considering all these mentioned pros, it can be stated that the researchers and organizations are the separate drive for more advanced models, which will further accelerate the development of NLP, and the rapid development of NLP is the result of this talented collaboration.

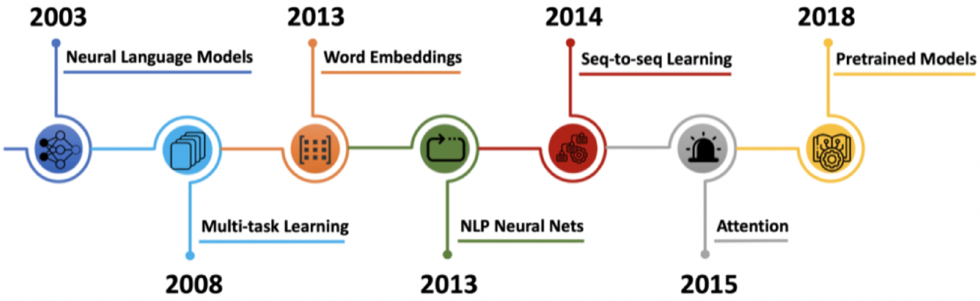


Figure 2.5: Evolution of NLP [2]

2.5.2 NLP Pipeline

Regardless of the target task in NLP studies, there are certain generally accepted methods that must be done before the data is used in the model to be developed. This processing pipeline usually involves a series of steps that transform and analyze text data to extract meaning and information. It provides standardization and consistency to data, ensuring reproducibility of results in research and model development. The procedure referred to as text preprocessing is a crucial component in the field of NLP. It plays a vital role in transforming unprocessed textual data into a structured and refined format suitable for subsequent analysis and modeling. First and foremost, it is common for raw text data to contain various forms of noise and redundant data. The text may contain irrelevant components such as special characters, punctuation marks, HTML tags, or other artifacts that are not relevant to the main objective at hand. Text preprocessing is the systematic removal or reduction of irrelevant components, leading to the creation of more precise and clear data. In addition, tokenization, which is a fundamental component of text preprocessing, contains the process of dividing the text into separate components known as words or tokens. This particular stage holds a fundamental role in various NLP tasks, as it serves as the fundamental building blocks for analysis. Tokenization serves as the initial stage in the processing of textual data for transformer models. Tokenization plays a crucial role in facilitating the development of vocabularies and word embeddings, which are necessary components for various NLP models. Text preprocessing aims to achieve standardization as a crucial objective. The conversion of all text to a consistent case, often lowercase, ensures the uniform handling of words, irrespective of their capitalization. The process of standardization is of crucial significance in the fields of text categorization, sentiment analysis, and information retrieval. Text preprocessing involves the elimination of "stop words," which are frequently occurring words such as "the," "and," or "in." These frequently used terms often have minimal impact on the overall semantic content and can be safely omitted, hence lowering the data's dimensionality. Dimensionality reduction techniques have been shown to improve the efficiency of models and reduce the impact of noise. Lemmatization and stemming are essential procedures that aim to reduce words to their fundamental or foundational form. An example of this is the simplification of words like as "running," "ran," and "runner" to it is lemma form, "run." This technique effectively captures the fundamental semantic content of words and reduces the total amount of the vocabulary. The management of special characters and numerical values constitutes an additional aspect of text preparation. The management of these variables may vary depending on the particular NLP task, requiring alignment with the research objectives. As an example, this might involve replacing numerical values with a symbol that is commonly used or effectively handling unique letters. Additionally, text preprocessing addresses the elimination of specific information, such as URLs, emails, or domain-specific jargon that may absence relevance to the analysis. After undergoing preprocessing, textual input must be transformed into numerical representations in order to be processed by machine learning models. Text is often converted into numerical vectors using methods such as one-hot encoding or word embeddings like Word2Vec or GloVe. In the case of transformer models, when the tokens have been inserted and location information has been implemented, a digitized representation of the text is obtained. The digitized tokens have been transformed into numerical representations, enabling the transformer model to effectively process them. Text preparation may involve addressing uneven data in certain cases. This holds particular significance in tasks such as sentiment analysis or text categorization, wherein certain classes may exhibit a lack of representation. To address this issue, various techniques such as oversampling, undersampling, or the utilization of class weights can be employed. Fig.2.6 can be given as an example of a basic NLP preprocess pipeline.

Following these steps, the process of building a model in line with the objectives of the relevant problem can be started. In addition to these steps, it is also possible to obtain useful information by applying various linguistic and statistical techniques, such as POS tagging, NER, and dependency parsing, to

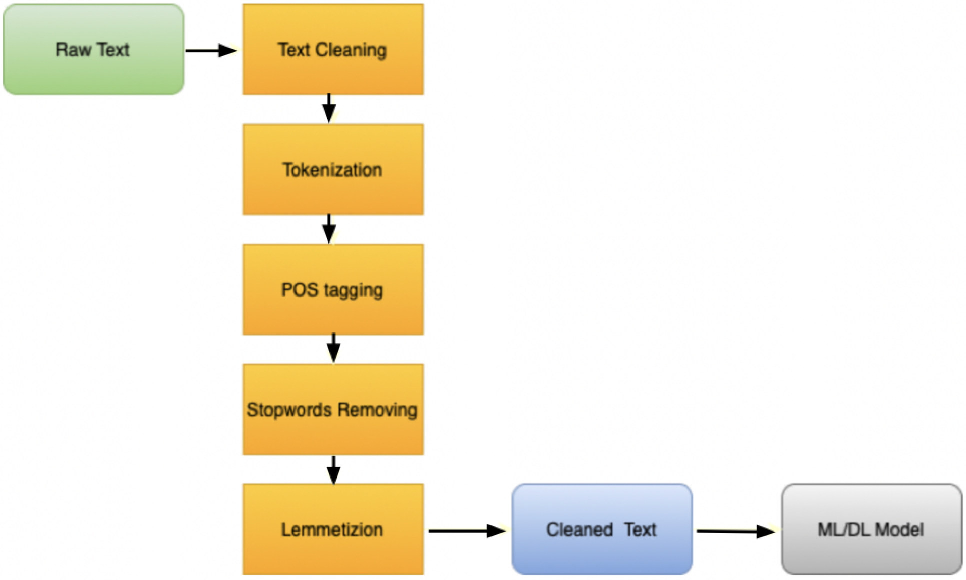


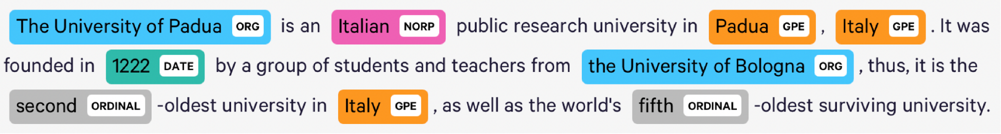
Figure 2.6: Example of a basic NLP preprocess pipeline

preprocessed text data. It is also possible to extract additional features that can be used for the model by applying feature extraction, which is one of the traditional methods. Examples of feature extraction methods include Bag of Words (BoW), TF-IDF, and word embedding. It is imperative to acknowledge that the aforementioned stages may exhibit variation and utilization contingent upon the objective of the pertinent issue and the attributes of the processed data.

2.5.3 NER

NER is a one of the NLP task that holds significant importance in the extraction of organized information from unstructured textual data. NER is a NLP technique that focuses on extracting important information from text, specifically referred to as named entities. These named entities might be single words, phrases, or even sequences of words. NER algorithms aim to recognize and classify these entities into specified categories. The text contains a wide variety of categories, with many themes including, in its simplest form, personal names, geographical locations, organization names, dates, events, and even exact values such as financial amounts and percentages. Furthermore, these categories have the potential to be organized and defined according to our customized problem. In summary, NER systems perform the task of processing textual data and identifying words or phrases that relate to things of significance. While performing this task, NER carries out the steps of Text Pre-Processing (tokenization), Entity Identification, Entity Classification and Contextual Analysis. NER plays a crucial role in various domains, encompassing but not limited to information retrieval, text summarization, question answering, and other related applications. As individuals, people possess the innate ability to effortlessly perceive and distinguish meanings and categories. This serves as evidence of our innate understanding of the surrounding environment. When it comes to computers, this seemingly straightforward task becomes a challenge wrapped with ambiguity. The aforementioned complexities emphasize the necessity of implementing a strong NER system, which involves instructing machines to comprehend diverse linguistic subtleties. To illustrate this intricacy with a specific instance: In the given statement, "Apple Inc. was established by Steve Jobs in Cupertino in 1976," a NER system would classify "Apple Inc." as an organizational entity, "Steve Jobs" as an individual entity, "Cupertino" as a geographical entity, and "1976" as a temporal entity. The utilization of this organized output can subsequently contribute to the improvement of text comprehension, the automation of data extraction processes, and the facilitation of diverse information retrieval endeavors. However, if one were to attempt information extraction using a rule-based approach rather than nNER, the system would encounter difficulty in determining the appropriate interpretation of the token "Apple" based on context, as it would be unable to discern whether it refers to the firm or the fruit.

The Fig.2.7 shows an example sentence separated into NER tags, taken from a spaCy[2] based online tool. Here, the nationality (norp), organization (org), geopolitical (gpe), ordinal and date entities in the relevant sentence are captured. It is important to acknowledge that the pertinent entity groups to be included display variability dependent on the specific challenge at hand.



The University of Padua ORG is an Italian NORP public research university in Padua GPE , Italy GPE . It was founded in 1222 DATE by a group of students and teachers from the University of Bologna ORG , thus, it is the second ORDINAL -oldest university in Italy GPE , as well as the world's fifth ORDINAL -oldest surviving university.

Figure 2.7: NER example

[2] https://spacy.io

Before moving on to the NER application, it is useful to mention POS tagging and IOB also commonly referred to as the Inside outside beginning (BIO) formats, which are key concepts. The method of POS tagging involves assigning labels to words in a given text based on their respective part of speech, which may include adjectives, verbs, or nouns. The main POS tagging is to clarify the syntactic and semantic functions of words in a sentence, offering valuable understanding of the grammatical organization of the text. The utilization of the BIO labeling format is crucial in dividing the limits of specified entities inside a given text. The tokens in this format are classified into three primary categories: B (Beginning), I (Inside), and O (Outside/Other). The letter "B" is used to indicate the initiation of a named entity, while the letter "I" is employed to indicate the continuation of an entity. On the other hand, the letter "O" is used to represent tokens that do not pertain to any named entity. The BIO format is designed to optimize the process of identifying named entities and aiding NER systems in effectively differentiating them from ordinary words. The framework offers a well-organized structure for delineating the extent of named entities, hence streamlining the process of extracting information.

NER systems commonly depend on machine learning algorithms and extensive datasets to train models capable of reliably identifying and categorizing items within textual input. These approaches can be rule-based, statistical, deep learning-based,transfer learning based, unsupervised or hybrid approaches and their performance depends on the quality of the training data and the level of challenge of the task.

The platform offers a wide range of tools and libraries that can be utilized for the implementation of NERtasks. There exist a multitude of pre-trained models that can be employed for NER, with notable examples including spaCy, Stanford NER[3], and NLTK[4]. spaCy is a free open source library that is utilized for NLP operations within the Python programming language. The NER system has various functionalities, including POS tagging, dependency parsing, and word vectors. NLTK serves as a framework for developing Python applications that manipulate and analyze textual data in human language. While the platform

[3] https://nlp.stanford.edu/software/CRF-NER.shtml

[4] https://www.nltk.org

is mainly recognized and widely used for its proficiency in analyzing linguistic data, it can also be employed for NER. Stanford University's NLP Group provides a diverse array of technologies designed for the purpose of NLP. One notable characteristic of the system is RegexNER, a rule-based interface that is specifically tailored for NER utilizing regular expressions. In addition, there are other API access options that can be utilized, including Google Cloud NLP[5], IBM Watson NLU[6], and OpenAI GPT-4 API[7]. When making a decision between constructing a bespoke NER model or utilizing an API, it is crucial to take into account one's individual requirements and the level of sensitivity of the data involved. Custom models provide a higher degree of flexibility when it comes to handling specialist jobs and also offer enhanced data privacy measures. However, it is important to note that the development and implementation of custom models necessitate a greater allocation of resources. On the contrary, APIs exhibit notable efficiency and affordability when employed for common tasks, however they may fall short in satisfying particular demands.

2.5.4 Transfer Learning for NER

As stated in the previous section, numerous techniques are employed for the implementation of NER. Transfer learning has become a popular approach in the field of NLP in recent times. This approach enables models to utilize pre-existing knowledge gained from prior training to enhance their performance on specific tasks. This is in addition to the conventional approaches employed in NLP. The objective of this study is to enhance the effectiveness and efficiency of classic NER methods by utilizing pre-trained contextual embeddings offered by BERT or GPT models. This approach intends to overcome the issues commonly encountered in traditional NER approaches. In this clear explanation, the key elements of Transfer Learning will be briefly highlighted. Transfer learning improves the efficiency of data utilization by making use of pretrained models that have been trained on large datasets. This approach shows to be particularly beneficial in cases where there is a limitation of labeled data available for a certain NER task. Pretrained models, such as BERT and GPT, are capable of capturing broad patterns in language. The process of fine-tuning a model using task-specific data

[5] https://cloud.google.com/natural-language

[6] https://www.ibm.com/products/natural-language-understanding

[7] https://openai.com/blog/gpt-4-api-general-availability

improves its capacity to understand the data, resulting in improved performance in NER compared to training the model from scratch. Moreover, the process of training NER models from the beginning can incur significant computational costs. Transfer learning addresses this issue by offering a pre-existing foundation of acquired representations, hence diminishing the need for extensive training time and resource allocation. Transfer learning is a technique that enables the adaptation of models to different domains, hence enabling them to specialize in specific industries or topics. The model's capacity to extract entities within a certain context is improved by first undergoing pretraining on a range of varied datasets, followed by fine-tuning using data that is relevant to the domain. By using this approach, it becomes feasible to acquire customized models. It might be likened to providing the model with a tailored training session to enhance its performance in a particular domain.

2.5.5 BERT

BERT, a remarkable pretrained model, had training on massive corpora, which covered sections of the internet, in order to get a comprehensive understanding of the complexities inherent in language. The existing knowledge, which is stored in the model's parameters, can be effectively applied to subsequent jobs that have an insufficiency of labeled data. BERT is available in different sizes, with the terms "base" and "large" indicating the different model sizes. The key distinctions between BERT-base and BERT-large lie in the size of the model architecture and the number of parameters employed. The increased number of layers and parameters in BERT big facilitates the acquisition of complex patterns and correlations within the data. However, this enhancement necessitates greater processing resources. BERT, with its bidirectional attention mechanism, showed an unparalleled capacity to perceive the context and nuances of language. In the era prior to BERT, a language model could get training by examining this textual sequence either from left-to-right or by combining left-to-right and right-to-left approaches. This one-way strategy has the potential to yield favorable outcomes in sentence formation. During the creation period, it is possible to anticipate and incorporate the subsequent word into the sequence, followed by the anticipation of the subsequent word, until a comprehensive phrase is formed. In contrast to the task of predicting the subsequent word in a given sequence, BERT employs an innovative method known as Masked Language Models (MLM).This strategy involves randomly masking specific words inside a sentence and subsequently attempting to predict them.In the context of this research, a "sentence" is defined as a random sequence of consecutive text, rather than a grammatically complete language sentence. The term "sequence" belongs to the input token sequence provided to BERT, which might consist of either a single sentence or two phrases combined. The initial token of each sequence is consistently marked as a distinct [CLS]. The combination of A and B sentence pairs is achieved by merging them into a unified sequence, with the usage of [SEP] tokens to mark their boundaries. This aids BERT in comprehending the boundary between the two sentences. MLM and Next Sentence Prediction (NSP), the methods BERT uses, can be briefly explained as follows:

1. MLM : The process of masking involves the model's ability to consider the entirety of a sentence, including both previous and succeeding words, in order to make predictions about the word that has been masked. In contrast to previous language models, this particular model takes into account both previous and succeeding tokens concurrently. The absence of this simultaneous part was not present in the preexisting models that blended left-to-right and right-to-left Long Short Term Memory (LSTM) architectures. The underlying concept at hand can be briefly stated as follows: a percentage of 15% of the words within the provided input are subject to random masking, wherein they are replaced with a "[MASK]" token. The complete sequence is processed using the BERT attention-based encoder, after which only the masked words are anticipated to predict. Context provided by other unmasked words in the sequence. Nevertheless, a drawback arises from employing a pure masking strategy. Although the model just focuses on predicting the presence of the "[MASK]" token in the input, our objective is for the model to accurately predict the correct tokens regardless of the specific token present in the input. In order to handle this problem, it is seen that out of the total tokens picked for masking, approximately 15%, a significant percentage of 80% will be replaced with the token "[MASK]". Approximately 10% of time tokens undergo replacement with a randomly selected token. A total of 10% of time markers remain unaltered. In the training phase, the loss function of BERT focuses mainly on the prediction of masked tokens, while ignoring the prediction of unmasked tokens.

2. NSP : The BERT training technique involves next sentence prediction as a means for learning the relationship between two sentences. The intended use of a pretrained model with this type of understanding relates to tasks such as question answering. During the training process, the model is provided with pairs of sentences as input. The objective of the model is to learn and predict whether the following sentence in each pair is the subsequent sentence in the original text. The BERT model utilizes a distinct "[SEP]" token to indicate sentence boundaries. During the training process, the model is presented with pairs of input sentences, where in 50% of

> the cases, the second sentence follows the first sentence. Approximately half of the instances involve the selection of a random sentence from the entire corpus. BERT is subsequently tasked with predicting if the second sentence exhibits randomness, on the assumption that the random statement will lack coherence with the first sentence. In order to determine the coherence between the first and second sentences, the entire input sequence is processed by a Transformer-based model. The resulting output of the "[CLS]" token is then passed through a classification layer, resulting in a vector with dimensions of 2x1. The IsNext-Label is assigned using the softmax function. The training process involves the simultaneous utilization of both MLM and NSP techniques. The objective is to minimize the combined loss function of the two techniques. Although the task is simple, previous studies have demonstrated the significant utility of pretraining in both question answering and natural language inference [11].

The comprehensive operational concept outlined is succinctly stated in Fig.2.8. The visual representation depicts the process of pretraining and fine-tuning in BERT. The symbol "E represents the input embedding, while "[CLS]" and "[SEP]" tokens are specialized symbols used for classification and sentence separation purposes, respectively. The symbol "C" is employed to denote NSP, while the symbol "T" is utilized to represent the final hidden vector for the ith input character in the sequence. Multispecies Natural Language Inference (MNLI), NER, and Stanford Question Answering Dataset (SQuAD) are widely recognized NLP tasks that have been extensively utilized for fine-tuning the BERT model. It is appropriate to categorize the symbols and tokens referenced in this context into three primary metadata categories. "[CLS]" token and "[SEP]" token as Token Embeddings, a marker indicating sentence "A" or sentence "B" as Segment Embeddings and Positional Embedding is added to each token to indicate its position in the sentence.

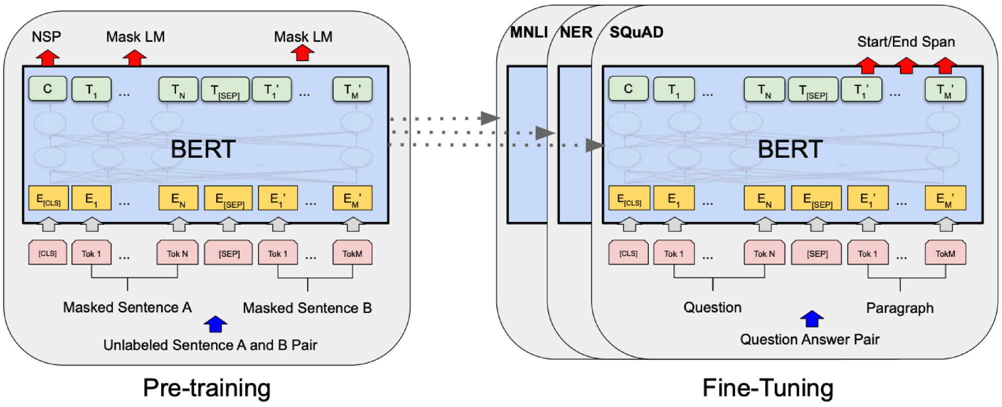


Figure 2.8: Main pre-training and fine-tuning procedures used for BERT [11]

In a nutshell, the utilization of a bidirectional strategy in BERT facilitated the acquisition of comprehensive contextual information, resulting in significant advancements in numerous NLP tasks, including question answering, sentiment analysis, and NER.

To understand how BERT uses all these components and the innovation they provide, we need to look at how they work and for this we dive into the Transformer architecture. The fundamental nature of a Transformer lies in it's neural network structure, which demonstrates exceptional proficiency in capturing the contextual links among words within a provided textual context [30]. In general, a fundamental Transformer architecture consists of two main components: an encoder and a decoder. These components are specifically intended to process incoming text and produce predictions relevant to a certain goal. However, in the case of BERT, the focus is on generating an extensive language representation model, and solely the encoder is utilized. The main essence of a Transformer resides in the neural network structure, which shows exceptional proficiency in capturing the contextual connections among words within a provided textual context. In general, a fundamental Transformer architecture consists of two components: an encoder and a decoder. These components are specifically intended to process incoming text and produce predictions relevant to a certain task. The encoder-decoder architecture employed in Transformers utilizes self-attention mechanisms, feedforward networks, and positional encodings to effectively process and create sequences. The encoder is responsible for processing the input sequence and converting it into a sequence of hidden representations. Subsequently, the decoder utilizes these representations to construct the output sequence. The initial step is the embedding of the input tokens, so generating a numerical representation. Positional encodings are incorporated into the model architecture to effectively capture and represent the sequential order of the input data. The encoder uses a multi-head self-attention method, facilitating tokens to pay close attention to other tokens with varying weights. This facilitates the development of a nuanced comprehension of the links present within the input sequence. However, in the situation of BERT, which aims to generate a comprehensive language representation model, only the encoder component is applied. The attention mechanism which is previously mentioned in the Evolution of NLP section, one of the important points of NLP, a distinctive feature of the Transformer architecture, is a key element in BERT's power to distinguish and capture complex contextual dependencies within the input sequence. The capacity to see and capture complex contextual interdependencies within the given sequence of input. The aforementioned technique allows the model to effectively concentrate on different segments of the input sequence, assigning distinct levels of significance to individual tokens based on their contextual relevance. Through this process, BERT is able to comprehend complex relationships between words, taking into account not just their immediate adjacent terms but also those located further away. As a result, BERT facilitates a comprehensive comprehension of language by considering a broader context. During the encoding process, the input tokens are sent through the encoder. In this stage, the attention mechanism plays a crucial role by dynamically assigning weights to each token based on its relative importance compared to others. This enables BERT to generate comprehensive and contextually aware representations. The encoding process, which is driven by attention mechanisms, plays a significant role in enhancing the model's exceptional language comprehension capabilities. It effectively captures the intricacies of syntax and semantics, outperforming conventional NLP methods in terms of complexity. The Fig.2.9 encompasses a comprehensive depiction of the entirety of this architectural structure. The overall design of the Transformer is characterized by the utilization of layered self-attention and point-wise, fully connected layers for both the encoder and decoder components. In essence, the effectiveness of BERT as a language representation model arises from the collaborative interaction between the attention mechanism and the encoder component inside the Transformer design. The encoder is responsible for transforming sequences of tokens into vectors that carry semantic meaning. In parallel, the attention mechanism enhances this encoding process by dynamically assigning different levels of importance to distinct tokens, depending on their contextual relevance. The complex interaction between several components enables BERT to demonstrate exceptional performance across a wide array of NLP tasks, establishing it as a fundamental element in the progression of comprehension and analysis of human language.

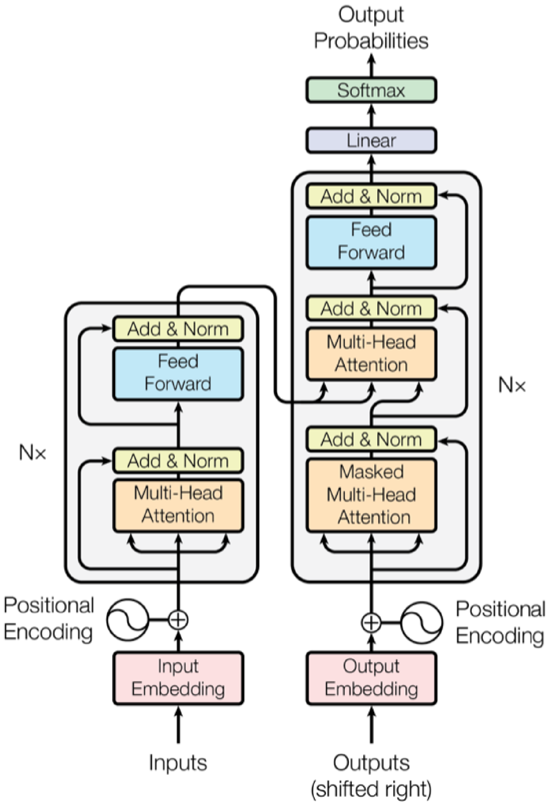


Figure 2.9: Model architecture of the Transformer [30]

3

EXPERIMENTS AND ANALYSIS

3.1 Dataset Generation

This section will thoroughly cover the process for generating the data collection, which is considered to be one of the most crucial aspects in the fine-tuning of BERT. In order to construct the model, it was necessary to generate a corpus as there was an absence of a designated labeled or unlabeled dataset pertaining to BMS. In this context, the first phase of assembling the dataset, with particular emphasis on patent documents related to BMS, for the goal of constructing a corpus, is examined. The initial phase of collecting these documents establishes the foundation for further examination of the language patterns and contextual complexities inherent in this specific set of texts. It then details the pre-processing steps performed to clean and prepare the dataset, ensuring that subsequent analysis is based on high-quality, standardized information. The efficacy of BERT is highly contingent upon the quality of the input data, rendering this step pivotal in achieving successful outcomes from the NER perspective. The latter section of this study focuses on the creation of token lookup files and the process of labeling the dataset. The utilization of token search files enables the optimization of tokenization processes, while the inclusion of dataset labeling is important in order to effectively train the NER model to identify and extract pertinent attributes from battery data.

3.1.1 Collection of BMS Patent Documents

An available dataset suitable for the this study is currently absent. Due to the lack of accessible open source RE documents in the BMS field and the limited number of sample RE documents available, patent documents in the BMS field were collected while creating the dataset due to the similarity of RE and patent documents, as previously mentioned in chapter 2. During the process of document collection, the selection was conducted based on many variables including the appropriateness of the domain name, the quality of the data, the structure of the documents, the variety of the data, and the substance of the documents. The selection process prioritized patents pertaining specifically to BMS, as opposed to more broad inventions related to battery, in order to ensure the appropriateness of the domain name. The study aimed to assess the quality of data and document structure in patent documents, with a focus on enhancing readability and cleanliness. The readability of patent documents is crucial for Optical Character Recognition (OCR) analysis due to the fact that these documents are typically viewed in image format rather than text format. OCR technology is employed to examine a scanned picture or document and transform it into text that can be interpreted by machines. Visual information may be found in patents as a result of their inherent characteristics. Given the primary emphasis of our study on the textual material inside patent documents, we specifically directed our attention towards patent documents that possess a substantial amount of text, as opposed to those that primarily consist of visual elements. This deliberate selection was made with the intention of minimizing the potential loss of information that may occur when images are excluded during the preprocessing stage. In relation to the variety of data, it is common practice for patent documents to be issued in two distinct standards, namely European and United States. In light of this justification, possible differences in sentence patterns and explanation were taken into account in the selection of patent documents published in both standards. The content quality of the claim section has been a significant factor in the selection of patents, as it has the closest resemblance to RE documents inside patent structures. After careful consideration of several pre-filters, it was determined that a total of nine patent documents were deemed appropriate for inclusion in the research. Out of the total number of patents, six is utilized as training data for the model, while three were subsequently set aside for evaluating the model's accuracy measures. The reasons that motivated this division will be discussed in the section 3.3. In addition, all preprocessing operations applied on the data described in the following sections will be the same for training and test data.

3.1.2 Applying Preprocessing Steps on Dataset

Converting Scanned Patent Documents to Text Format

The utilization of online OCR software has been recommended as a solution to address the challenge of dealing with patent PDF documents, which typically appear as pictures rather than clearly readable text, preventing direct information extraction. Various conventional OCR techniques, such as the popular Pytesseract[1] library and other OCR libraries like EacyOCR [2], have been employed in attempts to transform these pictures into textual format. Nevertheless, the absence of a standardized format for patent documents and the inclusion of a two-column layout in certain patent documents have introduced intricacies to the OCR extraction procedure. The efficacy of OCR libraries was impeded by the intricate layout of two-column pages, leading to unsatisfactory output. Alternative methods were researched to ensure the accuracy and completeness of this research. Given the requirement for the development of an OCR code capable of dynamically processing documents based on their individual structures, and need for testing processes, an online OCR editor[3] were considered as a viable option. These tools, which can be accessed via web browsers, provide a user friendly interface for translating text that is embedded inside images into material that can be edited and searched. In contrast to conventional OCR libraries, online editors frequently include sophisticated algorithms capable of effectively processing intricate document layouts, including the prevalent double-column format found in patent documents. The implementation of this particular feature has led to the identification of a resolution for one of the preexisting challenges. Considering the count of data, the OCR process to be performed via online tools will be less time consuming than an OCR code developed at this level, so the data was converted into text format in this manner.

[1] https://pypi.org/project/pytesseract/

[2] https://pypi.org/project/easyocr/

[3] https://www.pdf2go.com/pdf-to-text

NLP Preprocessing Steps

As previously discussed, the implementation of data preprocessing, a crucial stage in NLP research, involves subjecting raw data to specific processes. Given the nature of the data utilized in the research and the specific objectives of the study, it was necessary to perform the following preprocessing steps. Numerical patterns in the form "[0000]", which were used especially when listing claims or for itemizing the content, were removed and replaced with newline characters. Characters, which are used to represent hexedecimal values that can be frequently encountered in texts, have been removed. Only the "-" space character between letters or spaces is replaced by a space character. The numbers between them were not removed because they could indicate a date or a range of values and this information was thought to be important. Multiple characters representing spaces and newlines have been reduced to one. Additionally, other special characters, except the dot, have been eliminated. The dot character is kept for the next step of separating the text into sentences. The abbreviations like "fig.", "figs.","etc.",'e.g.' are commonly used in text. Frequently used abbreviations, such as "e.g." (for example), "i.e." (that is), and "etc." (et cetera), are expanded to their complete forms, such as "for example," "that is," and "et cetera," respectively. Custom block words specified in an external file called "custom_sw_list.txt" are excluded from the text. This step helps eliminate common words that may not contribute significantly to the NLP task. Natural Language Toolkit (NLTK)[4] is a library widely used in NLP and computational linguistics research and education that contains a suite of text processing libraries for classification, tokenization, source separation, tagging, parsing, and more. NLTK's sentence specifier is applied to separate text into individual sentences. This step is crucial for breaking the text into meaningful units and aiding subsequent word level tokenization. Finally, each sentence is converted into words using NLTK's word specifier, creating a word-level structured representation of the text. As stated before, a token with "[CLS]" and "[SEP]" tokens was added at the beginning and end of the sentences. While keeping the sentences at the word level provides convenience for the subsequent data labeling step, it is also necessary for the format of the data to be used for the BERT fine-tuning phase, which will be mentioned later. In conclusion, the aforementioned preparation

[4] https://www.nltk.org

procedures together enhance the quality of the initial text input, rendering it more appropriate for fine-tuning the BERT-NER model. The dataset has been enhanced with sentences that are well structured and words that have been tokenized. This serves as the foundation for following processes such as feature extraction and model training. The model's effectiveness in detecting named things in the provided text may be enhanced by careful evaluation of numbers, abbreviations, and words to block. This involves focusing on key linguistic patterns and excluding irrelevant information.

3.1.3 Creation of Token Lookup Files and Dataset Labeling

During the process of feature extraction from BMS safety RE documents, a decision was made to gather the relevant, commonly used, and technically valuable characteristics into four main groups at the start of the project. The main named entity categories for the custom BERT model under development are safety terminology, event, conditions and digital signal state aligning with the intended objectives. Each group contains NEs that can consist of a single word or several word groups. These groups are kept as a list in a .txt file named after them.

Category of NER Tags	**NER Tags**	**Sample**
Safety Terminology	B-SAFETY, I-SAFETY	harm, risk, safe state
Event	B-EVENT, I-EVENT	start, button press, timeout
Conditions	B-COND, I-COND	overdischarge, under voltage
Digital Signal State	B-DSS, I-DSS	high, low, absent

Table 3.1: Custom categories of NER tags identified

The BIO structure serves as a basis for labeling the dataset throughout the process of data annotation. In the process of NER annotation, when a token or a sequence of tokens is identified within the lookup.txt files, it is assigned a label based on the name of the .txt file in which the token(s) are present. In the event that the terms "risk" or "functional safety" are identified inside the safety.txt file associated with the safety terminology group, they must be labeled as "B-SAFETY". In the case where the NER description has several words, the subsequent word will be labeled as "I-SAFETY". In the event that the word in question cannot be located inside any of the lookup.txt files, it will be labeled as "other" and defined the "O" label. The previously mentioned procedures are executed in accordance with a rule-based approach, utilizing Python programming language for code development. Regular expression (regex) functions have been integrated into the NER labeling process in order to address instances when words do not exhibit an exact match one another due to variations in their forms and potential mismatches. For instance, in the event that the phrase "risks" was found instead of "risk", the system employed regular expression routines to identify and label such occurrences accordingly. This enhanced method is motivated by the approaches frequently employed in NLP research. It allows for a more extensive and precise identification of entities in BMS security documents. Within this particular framework, the NER tags that have been properly produced for use as labels in the future NER model are specified in Table 3.2, together with each of their descriptions and count.

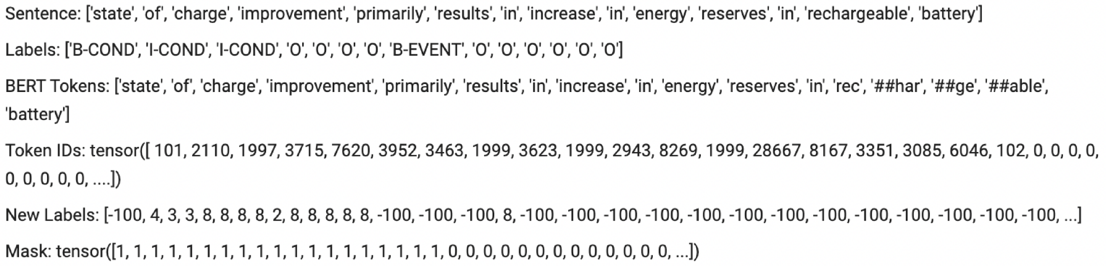
NER Tag	**Description**	**Count**
B-SAFETY	Beginning of a safety NEs	254
I-SAFETY	Inside a safety NEs	6
B-EVENT	Beginning of an event NEs	46
I-EVENT	Inside an event NEs	4
B-COND	Beginning of a condition NEs	454
I-COND	Inside a condition NEs	358
B-DSS	Beginning of a digital signal state NEs	172
I-DSS	Inside a digital signal state NEs	9
O	Tokens that are not identified as any NEs	43008

Table 3.2: NER tags and their counts in train dataset

3.2 Input Formatting for BERT Model Fine-tuning

The BERT model, which has been pretrained, has acquired valuable linguistic representations. Fine-tuning enables the adaptation of these representations to the specific task at hand, NER. To maintain compatibility, it is necessary to employ the identical architecture and tokenizer utilized during pretraining when fine-tuning a pretrained BERT model for a downstream task. The pretrained BERT-base model and its tokenizer were imported from the Hugging

Face model center[5] using pyTorch[6] due to this justification. In the preprocessing stage, the dataset has already undergone tokenization and is provided in the form of a pre-split list of words. In order to effectively integrate with the BERT corpus and handle words that are not available in this lexical resource, the BERT Tokenizer further breaks them down into subwords. In addition to tokenization, an essential phase in the process involves the use of padding to enable efficient batch processing of inputs. The main objective of this process is to provide a uniform length for input sequences within a given batch. In the context of BERT or similar models, which process inputs in batches, it is essential that all input sequences possess uniform length. Nevertheless, it is common for text data in real world scenarios to exhibit sentences of varying lengths. In order to achieve a consistent sentence length, a specific "[PAD]" token is intentionally inserted throughout the text [27]. The selection of the ideal length of padding is a complex and experimental procedure. The distribution of phrase lengths within the dataset is examined in order to provide an educated determination regarding a suitable number for maximum length called "max_len". This factor provides that the sentences with increased padding not only sustain consistency but also conform to the intrinsic attributes of the data, supporting an ideal balance between computing effectiveness and the retention of significant linguistic context.



```
Sentence: ['state', 'of', 'charge', 'improvement', 'primarily', 'results', 'in', 'increase', 'in', 'energy', 'reserves', 'in', 'rechargeable', 'battery']

Labels: ['B-COND', 'I-COND', 'I-COND', 'O', 'O', 'O', 'O', 'B-EVENT', 'O', 'O', 'O', 'O', 'O', 'O']

BERT Tokens: ['state', 'of', 'charge', 'improvement', 'primarily', 'results', 'in', 'increase', 'in', 'energy', 'reserves', 'in', 'rec', '##har', '##ge', '##able',
'battery']

Token IDs: tensor([ 101, 2110, 1997, 3715, 7620, 3952, 3463, 1999, 3623, 1999, 2943, 8269, 1999, 28667, 8167, 3351, 3085, 6046, 102, 0, 0, 0, 0,
0, 0, 0, 0, 0, ....])

New Labels: [-100, 4, 3, 3, 8, 8, 8, 8, 2, 8, 8, 8, 8, 8, -100, -100, -100, 8, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, -100, ...]

Mask: tensor([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...])
```

Figure 3.1: Input example formatted for BERT model

The Fig.3.1 illustrates the proper formatting phase of the syntax "state of charge improvement primarily results in increase in energy reserves in rechargeable

[5] https://huggingface.co/bert-base-cased

[6] https://pytorch.org

battery" for the BERT model. This syntax is a sample obtained from the training data and is provided as input to the model. A sentence refers to a unit of text that has been tokenized and then included in a list. Labels refer to the matching labels assigned to each word or token inside a given phrase. Each tag indicates whether the word belongs to an entity and specifies the type of entity. The initial statement is subjected to tokenization and processed by BERT using the BERT tokenizer. It is then represented as a sequence of BERT tokens. The token IDs are obtained by converting BERT tokens into numeric IDs using the dictionary specific to BERT. The input utilized by BERT consists of a numerical representation. New labels, these are modified labels used during training. "-100" labels are used to mask certain tokens during training when BERT processes fixed-length sequences. Moreover, there is a binary mask that denotes the elements in the array as either tokens (1) or fillers (0). This mechanism aids BERT in discerning between authentic tokens and padded token. In conclusion, Fig.3.1 has been produced for the purpose of training a NER model with BERT. The training process of the model involves utilizing tokenized and digitized input, implementing attention mechanisms, and making predictions of entity labels for each token in the given array. The purpose of the mask is to facilitate the model's ability to disregard padding tokens throughout the training process.

Before proceeding to the fine-tuning step, the data is split into two parts: train and validation. The utilization of a separate validation set allows for the evaluation of the model's efficacy on previously unknown data during the training phase. The purpose of the validation set is to serve as a diagnostic tool, mitigating the risk of overfitting, a phenomenon in which the model gets too specialized to the training data and hence fails to effectively generalize to new, unseen instances. The careful utilization of a validation set prior to fine-tuning guarantees that the model is refined to attain optimal performance on both the training data and unseen data, ultimately resulting in a more resilient and dependable BERT-based NER model. Based on the provided information, the dataset has been carefully split into two parts. Specifically, 10% of the dataset has been allocated for the validation set, while the remaining 90% has been assigned to the training set. Samples were randomly selected with pyTorch's "random_split" function, ensuring the split was representative of the overall dataset. After this process, 1,296 of the 1,441 sentences allocated for training were used for training the sample model, while 145 samples were reserved for validation.

In addition, the size of the test dataset, which is presented in the same way and will be utilized to measure the performance of the model, has been established as consisting of 637 test sentences.

3.2.1 BERT Model Fine-tuning

The text corpus including annotated NEs, which was suitably formatted in the preceding phase, was thereafter utilized as the input for the fine-tuning procedure. The adjustment of BERT settings was primarily aimed at enhancing the performance of NER. In each training iteration, a thorough traversal of the training set was performed, with a batch size of 32 that was carefully selected to enhance computing performance. The optimization procedure was executed by utilizing an AdamW optimizer, which was responsible for dynamically adjusting the parameters of the BERT model during the training process. While the learning rate of the optimizer is determined as 5e-5 the Epsilon value that prevents division by zero was determined as 1e-8. The training protocol consisted of a total of 10 epochs, with each epoch being a sequence of carefully performed procedures for every batch. The steps were executed sequentially. The gradients were reset to a value of zero. The model generated predictions, indicated as "preds". The difference between observed and predicted outcomes was calculated. The calculation of gradients was performed using the backpropagation algorithm. In order to address the well-known issue of bursting gradients, the gradients were subjected to clipping. The optimizer settings were modified by utilizing the gradients that were computed. After each epoch, the training loss was calculated, which provided an indication of the model's performance on the complete training set. The calculation of the average training loss gave a value of 0.01. In a coherent conclusion, the function effectively delivers both the average loss and the model predictions, so offering a thorough assessment of the model's competence and proficiency in accurately identifying complex patterns inside the annotated named entities.

3.3 Evaluation Metrics

When performing training for a machine learning model, such as the process of fine-tuning a model like BERT, it is of crucial significance to assess it's performance on a test set that has not been previously seen by the model. The purpose of this evaluation is to determine the model's ability to generalize effectively to new and previously unknown data. When the model is evaluated on the identical dataset employed for training or validation, the outcomes may exhibit a false sense of optimism. This is due to the possibility that the model could only memorize patterns within the training data, rather than genuinely acquiring the ability to generalize to new instances. The Accuracy measure, which quantifies the proportion of correctly predicted observations out of the total observations, is widely used in model evaluation. However, it's suitability may be limited in the context of imbalanced datasets, where one class is substantially more prevalent than the other. In addition to evaluating the efficacy of the model, it is important to thoroughly analyze the subsequent metrics [10].

1. The Precision refers to the level of accuracy or exactness in measurement or calculation. Precision can be defined as the proportion of accurately anticipated positive observations in relation to the overall number of expected positives. The metric pertains to the precision of positive forecasts. The precision formula is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

2. The Recall, also known as sensitivity or true positive rate, refers to the proportion of true positive instances that are correctly identified by a classification model. The concept of recall refers to the proportion of accurately anticipated positive observations in relation to the total number of actual positive instances. The metric quantifies the model's capacity to accurately represent and encompass all pertinent occurrences. The formula for calculating recall in a binary classification problem is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

3. The F1 Score is a commonly used metric in machine learning and statistics to evaluate the performance of a classification model. The method offers a trade-off between precision and recall, making it particularly advantageous in scenarios where there exists an imbalanced distribution of classes. The formula for calculating the F1 Score is derived as follows:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

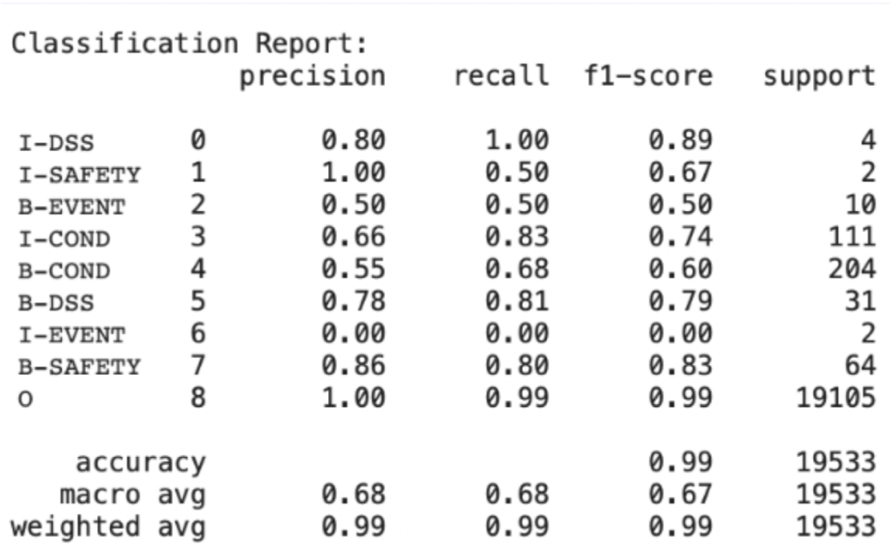
In the domain of NLP problems, similar to other domains within the realm of machine learning, it is crucial to include Precision, Recall, and F1 Score as fundamental metrics for evaluating the performance of a model. These metrics provide insights into the model's ability to accurately identify pertinent information while limiting the occurrence of both false positives and false negatives. When assessing a model's performance on a test set, it is desirable to achieve high accuracy and recall values, resulting in a high F1 Score. The F1 Score serves as an indicator of a favorable trade-off between precision and recall. These measures aid in making educated judgments on the performance and appropriateness of the model for the specific job at hand.

Additionally, the Confusion Matrix is a valuable method in evaluating the performance of a model. The matrix is useful for evaluating the performance of a classification algorithm across different classes. The confusion matrix offers valuable insights not just on the classifier's mistakes, but more significantly, regarding the specific categories of errors committed. The confusion matrix is a representation where the rows correspond to the actual classes and the columns correspond to the expected classes. It provides a detailed breakdown of the model's predictions by categorizing them into four outcomes: True Positives (correctly predicted positive instances), True Negatives (correctly predicted negative instances), False Positives (incorrectly predicted positive instances), and False Negatives (incorrectly predicted negative instances). From the confusion matrix, the previously mentioned Precision, Recall and Accuracy values can be calculated. These metrics, combined with the F1 Score, offer a comprehensive assessment of the model's performance and its suitability for specific tasks in NLP.

3.3.1 Analysis of Model Results

The Fig.3.2 presents a comprehensive depiction of the results obtained from the model. In addition to the metrics described in the previous section to measure the performance of the model, the support column indicates to the count of instances where the class is present inside the dataset that has been given. The metric represents the count of true instances for each category.

The model has remarkable performance in accurately classifying instances belonging to O/Other (class 8) NER tags, which is the predominant class within the dataset. It is crucial to ensure that the appropriate features requiring attention are not mistakenly attributed to unrelated entities. However, this finding



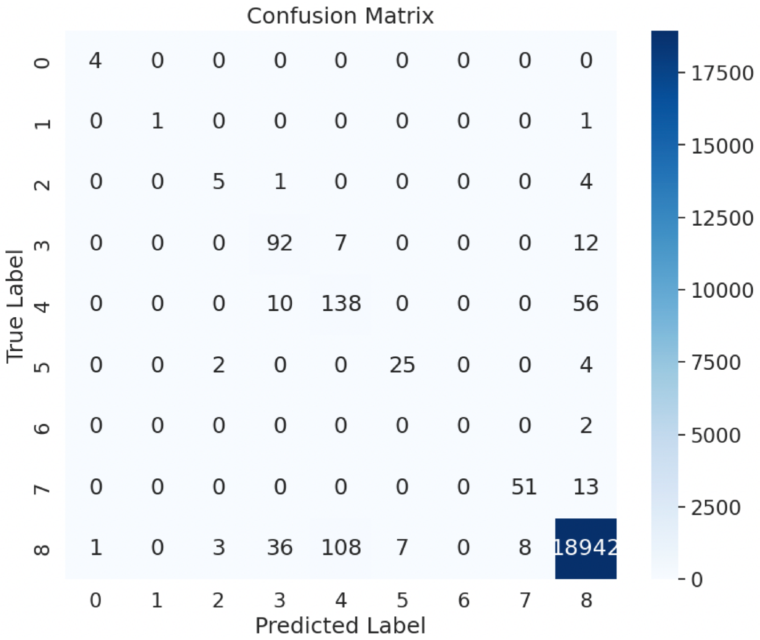
```
Classification Report:
              precision    recall  f1-score   support

I-DSS      0       0.80      1.00      0.89         4
I-SAFETY   1       1.00      0.50      0.67         2
B-EVENT    2       0.50      0.50      0.50        10
I-COND     3       0.66      0.83      0.74       111
B-COND     4       0.55      0.68      0.60       204
B-DSS      5       0.78      0.81      0.79        31
I-EVENT    6       0.00      0.00      0.00         2
B-SAFETY   7       0.86      0.80      0.83        64
O          8       1.00      0.99      0.99     19105

    accuracy                           0.99     19533
   macro avg       0.68      0.68      0.67     19533
weighted avg       0.99      0.99      0.99     19533
```

Figure 3.2: Classification report of the model

alone does not indicate that our methodology produces positive results. Furthermore, it is expected that the necessary NER classes will be accurately identified.

It is worth mentioning that B-COND, I-COND, B-DSS and B-SAFETY (classes 4, 3, 5, and 7) NER tags exhibit robust recall, and F1-Score, which signifies the model's proficiency in reliably detecting and categorizing occurrences belonging to these specific categories. Nevertheless, it is important to direct focus towards I-EVENT, I-SAFETY, and B-EVENT (classes 0, 1, and 2) NER tags since there exists opportunity for enhancement in terms of precision and recall. Specifically, I-EVENT (class 6) poses a significant issue as all metrics associated with it are recorded as 0.00. This indicates a clear requirement for targeted improvements in accurately identifying and categorizing instances belonging to this class. The assessment and mitigation of potential biases, class disparities, and problems related to generalization are of utmost importance in order to maintain the model's overall resilience across all categories. In general, the findings provide a strong basis, nevertheless, there is a need for focused enhancements to achieve optimal effectiveness, specifically in tackling certain obstacles encountered in the class.

Additionally, an examination of the confusion matrix depicted in Fig.3.3 allows for the identification of the classes that exhibit the highest levels of confusion with one another. As previously stated, the rows represent the true labels of the data, while the columns represent the projected labels predicted by the model. For instance, the test data contains a total of 204 instances belonging



	0	1	2	3	4	5	6	7	8
0	4	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1
2	0	0	5	1	0	0	0	0	4
3	0	0	0	92	7	0	0	0	12
4	0	0	0	10	138	0	0	0	56
5	0	0	2	0	0	25	0	0	4
6	0	0	0	0	0	0	0	0	2
7	0	0	0	0	0	0	0	51	13
8	1	0	3	36	108	7	0	8	18942

Figure 3.3: Confusion matrix of the model result

to the B-COND NER tag class (class 4). Out of the total 204 samples, 138 were accurately identified, while 10 were assigned the prediction of I-COND NER tag (class 3) and 56 were assigned the prediction of Other (class 8). Similarly, 92 samples of the 111 samples belonging to I-COND NER tag (class 3) were classified correctly, 7 samples were predicted as B-COND NER tag (class 4) and 12 were predicted as Other (class 8). This suggests that the model may have difficulty distinguishing between these closely related classes. Other (class 8) seems to be a common target for misclassifications from other classes. This might indicate that instances that are difficult to classify often end up in it.

In conclusion, the model performs well for some classes but struggles with others, particularly I-EVENT NER tag (class 6). Further analysis, feature engineering, or model adjustments may be needed to improve performance, especially for classes with lower F1 scores. Understanding the specific patterns of misclassifications can guide targeted improvements in the model.

4 CONCLUSIONS AND FUTURE WORKS

Evaluation of the BERT-based model for NER in the context of battery data extraction has yielded meaningful results. The classification report shown and analyzed in the Fig. 3.2 provides a comprehensive overview of the model's performance on various NER classes. The results show that an approach involving fine-tuning BERT on a battery-specific corpus is feasible, leveraging its capacity to learn representations from large-scale text data. As the initial stage of a system that makes a significant contribution to accelerating and automating RE document review, by offering a contextually sensitive approach to feature extraction through the use of BERT-based architectures in projects developed in the BMS field, with models to be improved and extended. The current results appears to hold potential for positive outcomes. Considering future work, in the pursuit of enhancing the use of BERT for feature extraction in battery data, an important way for progress is enhancing and expanding the existing model by including a more comprehensive and larger labeled dataset. The elimination of limitations related to labor and time resources is believed to facilitate the creation of a more dependable user-labeled dataset. This dataset, which enables context-based learning of NER, is considered an exciting basis for future research, overcoming the limits of rule based labeled datasets. The utilization of a comprehensive and dependable annotated dataset would significantly enhance the acquisition of knowledge pertaining to the BMS field through the improvement of the BERT-NER model. By augmenting the current 4 main NER tag categories and incorporating more categories for additional labeled entities, a more comprehensive comprehension of pertinent documents and enhanced feature extraction may be attained.

References

[1] Yara Alkhader, Amjad Hudaib, and Bassam Hammo. “Experimenting with extracting software requirements using NLP approach”. In: *2006 International Conference on Information and Automation*. IEEE. 2006, pp. 349–354.

[2] Antoine. “NetBERT: A Pre-trained Language Representation Model for Computer Networking”. In: 2020. URL: `https://api.semanticscholar.org/CorpusID:231658902`.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[4] Yoshua Bengio et al. “A Neural Probabilistic Language Model”. In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 1137–1155. ISSN: 1532-4435.

[5] Daria Berdyugina and Denis Cavallucci. “Automatic extraction of inventive information out of patent texts in support of manufacturing design studies using Natural Languages Processing”. In: *Journal of Intelligent Manufacturing* 34.5 (2023), pp. 2495–2509.

[6] Guillaume Brau, Jérôme Hugues, and Nicolas Navet. “Towards the systematic analysis of non-functional properties in Model-Based Engineering for real-time embedded systems”. In: *Science of Computer Programming* 156 (2018), pp. 1–20.

[7] K. R. Chowdhary. “Natural Language Processing”. In: *Fundamentals of Artificial Intelligence*. New Delhi: Springer India, 2020, pp. 603–649. ISBN: 978-81-322-3972-7. DOI: `10.1007/978-81-322-3972-7_19`. URL: `https://doi.org/10.1007/978-81-322-3972-7_19`.

[8] Ronan Collobert and Jason Weston. “A unified architecture for natural language processing: Deep neural networks with multitask learning”. In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 160–167.

[9] Djavan De Clercq et al. “Multi-label classification and interactive NLP-based visualization of electric vehicle patent data”. In: *World Patent Information* 58 (2019), p. 101903. ISSN: 0172-2190. DOI: `https://doi.org/10.1016/j.wpi.2019.101903`. URL: `https://www.sciencedirect.com/science/article/pii/S0172219018300851`.

[10] Leon Derczynski. “Complementarity, F-score, and NLP Evaluation”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. 2016, pp. 261–266.

[11] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[12] A Hariprasad. “Battery management system in electric vehicles”. In: (2020).

[13] Manal E. Helal. *Patent Mining by Extracting Functional Analysis Information Modelled As Graph Structure: A Patent Knowledge-base Collaborative Building Approach*. 2023. arXiv: `2305.00309 [cs.DB]`.

[14] Guntur Budi Herwanto, Gerald Quirchmayr, and A Min Tjoa. “A named entity recognition based approach for privacy requirements engineering”. In: *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE. 2021, pp. 406–411.

[15] Shu Huang and Jacqueline M Cole. “BatteryBERT: A pretrained language model for battery database enhancement”. In: *Journal of Chemical Information and Modeling* 62.24 (2022), pp. 6365–6377.

[16] Kamaljit Kaur and Parminder Kaur. “BERT-CNN: Improving BERT for Requirements Classification using CNN”. In: *Procedia Computer Science* 218 (2023), pp. 2604–2611.

[17] Yi Li et al. “Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review”. In: *Renewable and Sustainable Energy Reviews* 113 (2019), p. 109254. ISSN: 1364-0321. DOI: `https://doi.org/10.1016/j.rser.2019.109254`. URL: `https://www.sciencedirect.com/science/article/pii/S136403211930454X`.

[18] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL: `https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf`.

[19] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: `1301.3781 [cs.CL]`.

[20] Dhirendra Pandey, Ugrasen Suman, and A Kumar Ramani. “An effective requirement engineering process model for software development and requirements management”. In: *2010 International Conference on Advances in Recent Technologies in Communication and Computing*. IEEE. 2010, pp. 287–291.

[21] Sameer Paradkar. *Mastering non-functional requirements*. Packt Publishing Ltd, 2017.

[22] Klaus Pohl. *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.

[23] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018).

[24] Lev Ratinov and Dan Roth. “Design challenges and misconceptions in named entity recognition”. In: *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*. 2009, pp. 147–155.

[25] I. Sommerville. *Software Engineering*. Always learning. Pearson, 2016. ISBN: 9780133943030. URL: `https://books.google.com.tr/books?id=tW4VngEACAAJ`.

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: `1409.3215 [cs.CL]`.

[27] Archana Tikayat Ray et al. “aeroBERT-NER: Named-Entity Recognition for Aerospace Requirements Engineering using BERT”. In: *AIAA SCITECH 2023 Forum*. 2023, p. 2583.

[28] Amy J.C. Trappey et al. “Intelligent compilation of patent summaries using machine learning and natural language processing techniques”. In: *Advanced Engineering Informatics* 43 (2020), p. 101027. ISSN: 1474-0346. DOI: `https://doi.org/10.1016/j.aei.2019.101027`. URL: `https://www.sciencedirect.com/science/article/pii/S1474034619306007`.

[29] Erik Vanem et al. "Data-driven state of health modellingA review of state of the art and reflections on applications for maritime battery systems". In: *Journal of Energy Storage* 43 (2021), p. 103158. ISSN: 2352-152X. DOI: `https://doi.org/10.1016/j.est.2021.103158`. URL: `https://www.sciencedirect.com/science/article/pii/S2352152X21008598`.

[30] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[31] 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`.

[32] Joseph Weizenbaum. "ELIZAa Computer Program for the Study of Natural Language Communication between Man and Machine". In: *Commun. ACM* 9.1 (Jan. 1966), pp. 36–45. ISSN: 0001-0782. DOI: `10.1145/365153.365168`. URL: `https://doi.org/10.1145/365153.365168`.

[33] Rui Xiong, Linlin Li, and Jinpeng Tian. "Towards a smarter battery management system: A critical review on battery state of health monitoring methods". In: *Journal of Power Sources* 405 (2018), pp. 18–29. ISSN: 0378-7753. DOI: `https://doi.org/10.1016/j.jpowsour.2018.10.019`. URL: `https://www.sciencedirect.com/science/article/pii/S037877531831111X`.

[34] Jun Yang et al. "A BERT and Topic Model Based Approach to reviews Requirements Analysis". In: *2021 14th International Symposium on Computational Intelligence and Design (ISCID)*. 2021, pp. 387–392. DOI: `10.1109/ISCID52796.2021.00094`.

[35] Liping Zhao et al. "Natural language processing for requirements engineering: A systematic mapping study". In: *ACM Computing Surveys (CSUR)* 54.3 (2021), pp. 1–41.

[36] Didar Zowghi and Vincenzo Gervasi. "The Three Cs of requirements: consistency, completeness, and correctness". In: *International Workshop on Requirements Engineering: Foundations for Software Quality, Essen, Germany: Essener Informatik Beitiage*. 2002, pp. 155–164.