TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS

PROJECT NO.: THA079MSISE017

INFORMATION EXTRACTION FROM EMAIL USING BERT

BY

TIKA SAH

A REPORT

SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATICS AND INTELLIGENT SYSTEMS ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

KATHMANDU, NEPAL

AUGUST, 2024

INFORMATION EXTRACTION FROM EMAIL USING BERT

by

Tika Sah

THA079MSISE017

Project Supervisor

Er. Kiran Chandra Dahal

A mid term report submitted in partial fulfillment of the requirements for the degree of

Master of Science in Informatics and Intelligent Systems Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

Tribhuvan University

Kathmandu, Nepal

August, 2024

ACKNOWLEDGMENT

This project work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First of all, I would like to express my sincere gratitude to my supervisor, **Er. Kiran Chandra Dahal**, of **Head of Department , Thapathali Campus** for providing invaluable guidance, insightful comments, meticulous suggestions, and encouragement throughout the duration of this project work. My sincere thanks also goes to the M.Sc. coordinator, **Er. Dinesh Baniya Kshatri**, for coordinating the project works, providing astute criticism, and having inexhaustible patience.

I am thankful to my managing director at my company for providing insight into the idea of the project and assisting in finalizing the topic. My friend, along with the staff of my company who were involved in this project, deserves special mention for their collaborative efforts and support. I particularly want to acknowledge my managing director for agreeing to provide the necessary dataset from our work, which is crucial for the project.

I extend my heartfelt appreciation to the management and staff of my company for their invaluable assistance and commitment to this project title. Their collective efforts and dedication played a pivotal role in shaping the ideas and refining the content of this proposal. I am sincerely grateful for their guidance, encouragement, and willingness to provide the necessary resources to facilitate the progress of this endeavor.

I would like to express my gratitude to my colleagues and friends for their encouragement, constructive feedback, and moral support throughout the preparation of this proposal.

Tika Sah
THA079MSISE017
August, 2024

ABSTRACT

In the dynamic and highly competitive business environment of today, one's operational success relies on how fast and accurate the processing of every transaction is. Recently, EDI systems have swept the business world; however, a number of organizations still use emails for transactions due to various integration issues. It becomes very time-consuming to extract transactional data from received emails manually, and it is error-prone too, hence leading to inefficiencies. It creates an automated information extraction system using state-of-the-art NLP techniques, more particularly the BERT model. The system will be trained on a wide array of transactional emails for the extraction of key elements accurately, such as purchase ordersnotifications— from DOCX attachments. OurCelery-powered solution runs background tasks efficiently against unseen emails in real time. The structure of extracting data makes it fit seamlessly for the integration of downstream processes. Preliminary results flicker with significant accuracy improvements and efficiency enhancements, thus proving the scalability and flexibility of the system against various email volumes and formats. This project aims to revolutionize transaction processing by reducing manual effort in transactions to near zero and increasing data reliability.

Keywords: BERT Model, Celery , Information Extraction, Natural Language Processing, Transaction Processing .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AW-OIE	Automatic Wiki-OIE
BART	Bidirectional and Auto-Regressive Transformers
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
BILSTM	Bidirectional Long Short-Term Memory
CNNs	Convolutional Neural Networks
CRM	Customer Relationship Management
CRF	Conditional Random Field
EDI	Electronic Data Interchange
GUI	Graphical User Interface
GPT	Generative Pre-trained Transformer
IDE	Integrated Development Environment
IE	Information Extraction
LDA	Latent Dirichlet Allocation
MAP	Mean Average Precision
MLM	Masked Language Modeling
NAS	Network-Attached Storage
NER	Named Entity Recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
PMS	Predicate Matching Score
RNNs	Recurrent Neural Networks
W2V	Word2Vec

1 INTRODUCTION

1.1 Background

Computers have always found it struggle to understand human languages. Each comes with its own set of rules, grammar, and writing systems like the Devanagari script in Nepali or the Latin alphabet in English. Moreover, tasks such as finding certain details in an article or analyzing the whole speech add to the entire pretentiousness.Public data will also facilitate the opportunity for replication, comparison, and to build further on the work that has gone before. In this area, accessible data is legal due to its applicability to natural language processing. This domain currently relies on deep learning models, which are reliant on voluminous textual data for their training sets. Domain-specific datasets are, in general, required for NLP to benefit from the success of neural network techniques. As these demands for training data increase, such evolution continues with the neural language models' growth in both size and complexity. However, in certain situations, the sharing of data is limited by ethical and legal constraints. For example, in a medical situation, the General Data Protection Regulation does not give the public access to records that might be attributed to a certain individual. This is a process of removing or masking any identifiable information making them available for wider research, which includes names, addresses, birth dates, and social security numbers, from medical records. Automated anonymization is important since it is one of the ways that the labor-intensive process involved in manual anonymization can be solved. It is a difficult process to automate because the medical records are neither structured nor classified. This makes the problem of identification and categorization of sensitive information complex. These recent advances in deep learning for named-entity recognition even demonstrate promising prospects for applying NER techniques within the medical records anonymization process. However, computers could be made to retrieve very important information from articles and speeches, and even have discussions with the users, provided the right methodologies are used.

Traditional methods used by researchers to make sense of languages involved some intricate ways of breaking down the languages into smaller parts, such as words and sentences. This has since been changed by deep learning. Currently, powerful neural networks are driving natural language processing and giving computers the ability to

make sense of spoken and written languages. While for humans language acquisition is intuitive, for machines it still remains a complex endeavor.

Information Extraction has indeed made a significant contribution to the efficient organization of NLP with large textual data reams. Be it the bulk of information flowing through social media, detailed content of news articles, or stack of documents waiting to be analyzed, IE—powered by AI and NLP—can realize and retrieve relevant information automatically. It is applied to question-answering systems and management of social media content, where IE identities the patterns and extracts crucial information about names, locations, and dates. In this way, it saves not only time and resources in comparison with manual data entry but also reduces the possibility of errors. However, the effectiveness of IE depends on the text's specific requirements. The invoice extraction, for example, seeks company names and their billing addresses, whereas medical reports need access to the names of patients and data about medical medication. It is the knowledge of the types of data that are to be extracted and the use of appropriate tools in NLP that make IE a very effective tool for unlocking useful insights from the growing corpus of textual information.

BERT, Bidirectional Encoder Representations from Transformers, is a major game changer in NLP, proposed by Google in 2018. A lot of RNNs and CNNs had been in the NLP pipeline before the invention of BERT, processing data sequences in predefined orders, which seriously limited their scaling ability and efficiency. Then, in 2017, came the transformer model, which allowed the processing of data in any order; the path was paved for the coming of BERT. BERT, however, harnesses the power of the transformer to pretrain on enormous amounts of language data, thereby drastically improving NLP tasks.

The impact of BERT, however, is not witnessed in English alone but has far-reaching applications in more than 70 languages as of December 2019. Its broad implementation has facilitated search engine optimization and equally, voice and text-based search capabilities—something that had previously been hindered by inaccuracies of the NLP. BERT contextualizes and, therefore, arrives at an interpretation of a language pattern across different languages, smashing these linguistic barriers. Since then, BERT's success has helped to build lighter versions and train similar methods that have influenced

the subsequent AI systems like GPT-2 and ChatGPT. On the land of NLP and Artificial Intelligence, it will be continuously shaped by BERT, thereby evidencing the potential role innovated language models can play in furthering human-computer interaction and understanding.

NLP techniques, such as Named Entity Recognition and BERT models, can be very helpful in business contexts for processing and managing business emails. For example, these tools extract the key transactional elements from within the email and sometimes from DOCX attachments, like purchase orders, invoices, and shipment notifications. This paper presents a project on developing an automated information extraction system using BERT for transactional data from business emails. Integrating Celery for background tasks, the system will efficiently process unseen emails in real-time. Structuring the extracted data in a format that can be seamlessly integrated for further business processes, it will revolutionize the transaction-processing approach with respect to reduced manual effort, improved reliability of data, and better operational efficiency.

1.2 Motivation

The ever-growing mountain of emails is a major headache for organizations. As digital communication explodes, emails have become the go-to platform for businesses and individuals alike. But with this ease of use comes the struggle of managing and extracting valuable information from this massive sea of unstructured text.In the world of automation still extracting data from email is done manually which is laborious and error-prone. The motivation for this project stems from the prevalent use of Electronic Data Interchange (EDI) for automated transaction processing in business environments. EDI enables seamless electronic communication of transactional data between trading partners, streamlining processes and reducing manual intervention. However, many organizations still rely on email as a fallback mechanism for transactional communication, especially when EDI integration is not feasible or available. Despite the convenience of email, manually processing transactional information from email content can be time-consuming, error-prone, and inefficient. To address this challenge, the motivation for this project is to leverage advanced natural language processing (NLP) techniques, particularly the BERT (Bidirectional Encoder Representations from Transformers) model, to automate the extraction of transactional data from email content.

By training a BERT-based model on a corpus of email communications containing transactional information, we can develop an automated system capable of accurately identifying and extracting key transactional elements such as purchase orders, invoices, and shipment notifications. This automation not only enhances operational efficiency but also reduces the risk of errors and delays associated with manual transaction processing. Advancements in NLP, powered by cutting-edge models like BERT, offer a beacon of hope. These models mimic human thinking through deep learning, allowing them to analyze and extract information from unstructured text with impressive accuracy and speed.BERT shines in understanding context and meaning within text. Trained on massive amounts of data, it considers the entire email at once, leading to more precise interpretations of words and phrases. By harnessing the power BERT, organizations can automate information extraction from emails, saving precious time and resources. These models can pinpoint important entities, extract crucial details like dates and events, and even categorize emails based on their content. This allows organizations to streamline workflows, boost productivity, and gain valuable insights from their email exchanges.

1.3 Problem Definition

Despite the widespread adoption of Electronic Data Interchange (EDI) for automated transaction processing in business environments, many organizations still rely on email as a primary communication channel, particularly for transactional purposes. However, manually processing transactional information from email content can be timeconsuming, error-prone, and inefficient, leading to delays, inaccuracies, and operational challenges. This reliance on manual intervention hampers productivity, increases processing costs, and introduces the risk of errors and delays in transactional workflows. The problem at hand is the lack of an automated solution for extracting transactional data from email content effectively and accurately. Existing methods often involve manual data entry or ad-hoc parsing techniques, which are labor-intensive, prone to errors, and lack scalability. Moreover, the variability and complexity of email content, including nonstandard formatting, language ambiguity, and context-dependent information, further exacerbate the challenge of automating transaction processing from email communications.

1.4 Project Objectives

The primary objectives of this project are as follows:

- To develop an automated information extraction system utilizing the BERT model.
- To extract information from email content and attachments of type .doc, with the aim of automating the information extraction process.

1.5 Scope of Project

The project consists of the design and the implementation of an information extraction system for developing very-advanced techniques related to natural language processing, which will be targeted towards extracting relevant information from the email text. The proposed system will be based on state-of-the-art models such as BERT and task-specific additional layers on top for making relevant extractions. It can identify key entities, including PO numbers, item descriptions, and quantities, as accurately as possible from diverse and vague email communications. This project will basically include the work on a carefully extracted subset of business emails to perform evaluation, and it will have developed metrics which will include precision, recall, and F1 score apart from evaluation by users. It is envisaged that modifications and hyperparameter tuning with the models can be conducted in these low-resource settings to provide us with knowledge for future NLP systems in CRM, sales automation, market research, and compliance within any business organization. In using machine learning and natural language understanding, the system shall enable any business organization by smoothing information retrieving processes, improving the efficiency of work, and easing decision-making processes.

Some limitations may be faced by the automated information extraction system. These can be challenges of email content variability and complexity, ranging from nonstandard formatting to language ambiguity to context-dependent information. It is also possible that some emails might not yield correct extraction of the system with very unstructured or ambiguous text, thus requiring manual intervention or additional preprocessing steps. This could also be lixiviated by the quality and quantity of the training data used and how fine-tuning and model parameters are optimized. Thus, addressing these limitations is an important way forward if the system is to have successful deployment in real-world business applications and scale.

1.6 Potential Project Applications

1.6.1 Streamlining Business Operations

Little manual effort is needed when transactional data is automatically extracted from email communications for further processing. Along these very lines, organizational efficiency can be improved, which will aid in tasks such as order fulfillment, invoicing, and payment processing by automating transaction processing for better productivity. This would help the smooth running of business operations by ensuring proper allocation of resources to various tasks.

1.6.2 Financial Management

Automated extraction in financial management helps in straightforward, on-time, and accurate financial transactions with minimal errors and delays. This is very important to record-keeping because all transactions are accounted for and relevantly processed without unnecessary holdups. This could help in reconciliations, financial reporting, and financial compliance.

1.6.3 Customer Service

Faster responses to transaction-related enquiries and requests received via email can be automated for better customer satisfaction. Customer service quality can be offered with higher quality and increased loyalty by the customers since the system allows the customer service agents to quickly and precisely respond to the different type of enquiries by way of enquiry categorization and extraction of the relevant details.

1.6.4 Supply Chain Management

On transactional emails, automated processing will help optimize the process of order management and inventory control in supply chain management. Automated extraction and processing of information about orders, shipments, and inventory levels can be relied upon to perform supply chain operations efficiently and respond to demand changes promptly.

1.6.5 Smarter Email Filtering

The results of the project can make the whole process of filtering emails by basis of a project or urgency like never before. This can let any organization focus on the key communications and handle the email workflow effectively, making sure that each and

every important mail is taken seriously.

1.6.6 Automated Information Retrieval for Customer Support

Apart from developing the efficiency of customer support by way of categorization of customer queries and extracting relevant information, it makes quick and appropriate responses possible. In that way, the entire process makes customers more satisfied by reducing response times for responses, thus improving the quality of service and creating loyalty.

1.7 Originality of Project

- This project pioneers the use of BERT, a state-of-the-art NLP model, for the specific task of analyzing and extracting information from business email communications.
- Introduces an innovative solution to automate the extraction of transactional data from emails, addressing challenges related to efficiency, accuracy, and scalability in manual handling.
- To Accurately extract key transactional elements from content.

1.8 Organisation of Project Report

The project proposal has been worked as a complete roadmap, telling the reader very clearly and precisely, thereby leading them through its contents. It begins with the formal sections of the cover page and the title page, being professional in its sense. The abstract generalizes the proposal, summaries of the main points of the project being put forward and the purpose of the table of contents is to make it easy and time-efficient to navigate the document. Lists of figures, tables, and abbreviations give an additional idea about what content refers to and common terminology, which can be used in the report. The introduction section includes the background of the project, its objectives, limitations, potential applications, and distinguishing features. The literature review covers review of work related to this project. The methodology section includes theoretical background, mathematical models, system design, instrumentation requirements, and the methods of data collection and data analysis including human evaluation. Next are the results section, where the findings of the project are displayed, and the discussion and analysis section, which critically describes the results, bringing out what they mean; it discusses

possible limitations and their context in relation to the existing body of research. It further lists tasks awaiting completion in the project and appendices, which include additional materials such as the detailed project schedule and the recordings of the base paper. This structure ensures that the availability of information is easy and logical to the readers for understanding and evaluating the project scope and objectives.

2 LITERATURE REVIEW

In this section, the techniques which were used for NLP tasks in past will be explained in a detailed way. The techniques which were used are BILSTM, BERT (Bidirectional Encoder Representation from Transformers).

Extractive text summarization, encompassing both statistical and deep learning approaches. Statistical methods, such as frequency count-based (e.g., Luhn), graph-based (e.g., TextRank, LexRank), and latent semantic analysis (LSA), have been pivotal in the field. On the other hand, deep learning has ushered in significant advancements, including attentional encoder-decoder models like those proposed by Cheng and Lapata in 2016, autoencoder architectures (e.g., SummCoder), reinforce ent learning-based models (e.g., BanditSum), and BERT-based frameworks, exemplified by BERTSum introduced by Liu in 2019. The paper underscores the importance of robust evaluation metrics like ROUGE, which assesses the correspondence between generated summaries and human references. Specifically, it elaborates on BERTSum's methodology, highlighting its fine-tuning of pre-trained BERT models tailored for extractive summarization via input sequence and embedding modifications. Additionally, the paper introduces two novel models geared towards enhancing efficiency in extractive summarization: DistilBERTSum, which leverages knowledge distillation to compress BERT by 40% while retaining 97% performance, and SqueezeBERTSum, a model employing grouped convolutions to reduce parameters by 49% while maintaining 98% performance compared to BERTSum. Empirical findings showcase SqueezeBERTSum's competitive ROUGE scores despite substantially fewer parameters than its predecessor. Furthermore, the paper outlines prospective research avenues, including hyperparameter optimization, domain-specific summarization, exploration of abstractive summarization with SqueezeBERT, and the integration of model compression techniques like quantization and pruning.[1]

The authors proposed a novel hybrid model by combining BERT and BiLSTM architectures for sentiment analysis of Online Catchphrases and Buzzwords (OCBs). Leveraging BERT's pretraining language model, they utilized token embedding, segment embedding, and position embedding to capture contextual information effectively. Additionally, BiLSTM was employed to address long-term dependencies in the data, enhancing the

model's ability to understand and utilize intricate patterns within OCBs. The proposed model outperformed other methods in terms of F1-score, recall, and precision on the OCBs dataset, demonstrating its robustness in accurately classifying sentiment in informal and irregular text data. A comprehensive evaluation, including ablation experiments, validated the effectiveness of the proposed approach. However, the paper could benefit from a more detailed discussion on specific challenges in sentiment analysis of OCBs and comparisons with more recent models to further enhance its contribution to the field. [2]

BART (Bidirectional and Auto-Regressive Transformers) model, utilizing a denoising autoencoder approach for pretraining. They experiment with various noise reduction strategies, including rearranging phrase sequences and text infilling with mask tokens, within the BART framework. Quantitative results demonstrate improvements in performance metrics such as BLEU and ROUGE scores, showcasing the model's effectiveness in text summarization tasks. Qualitatively, the authors compare BART's performance with other prominent NLP models like RoBERTa and GPT, showing competitive or superior results. The paper's strengths lie in its clear methodology, comprehensive evaluation of noise reduction strategies, and alignment with current research trends. However, weaknesses include limited discussion on evaluation metrics' nuances, insufficient exploration of model limitations, and potential biases in experimental data selection. Addressing these weaknesses would enhance the paper's overall credibility and impact.[3]

A method for Open Relation Extraction (Open RE) that addresses the shortcomings of existing tagging schemes by introducing a novel tagging scheme capable of handling overlapping problems in sentences with multiple relations. Their proposed method employs a sophisticated model architecture comprising three layers: an embedding layer, a bidirectional LSTM layer, and a Conditional Random Field (CRF) layer for sequence tagging. By experimenting with various pre-trained word embeddings such as GloVe, Word2Vec, BERT, and ELMo, the authors demonstrate the versatility of their approach across different embedding methods. The training data is sourced from the AW-OIE dataset, while the models are evaluated on datasets including Wikipedia and NYT. Quan-

titative results showcase significant enhancements in recall and Predicate Matching Score (PMS) values when using the new tagging scheme, with the NTS-BERT-BiLSTM-CRF model achieving remarkable accuracy in extracting relations from sentences with multiple relations. Strengths of the paper lie in its innovative tagging scheme, advanced model architecture, and substantial performance improvements in relation extraction. However, the paper could benefit from a more extensive comparison with other state-of-the-art Open RE methods and exploration on a broader range of datasets to assess generalizability. Additionally, the computational complexity associated with utilizing advanced models like BERT and ELMo may hinder scalability in resource-constrained environments, warranting further investigation into optimization techniques. Overall, the paper presents a promising approach for Open RE, but further validation and refinement are needed to strengthen its applicability and credibility in the field.[4]

Investigation into the application of Natural Language Processing (NLP) techniques for knowledge discovery and information extraction within the field of energetics. Three distinct NLP models – Latent Dirichlet Allocation (LDA), Word2Vec (W2V), and Transformer – are meticulously trained and evaluated on a curated corpus comprising over 80,000 energetic documents. Each model undergoes specific training procedures tailored to its architecture, followed by comprehensive evaluation to assess its ability to capture and interpret energetics-related concepts. The LDA model, despite achieving a perplexity score indicating coherence in identifying topics, exhibits areas for improvement in interpretability. Conversely, the W2V model effectively clusters thematically similar words, demonstrating a strong alignment with energetic motifs. The Transformer model, fine-tuned on a domain-specific corpus, successfully predicts masked words characteristic of energetics documents. While each model showcases strengths in capturing critical energetics concepts, such as detonation thermodynamics and material formulation engineering, there are limitations, such as the need for further interpretability enhancements and the challenge of computational constraints in hyperparameter tuning for the Transformer model. Overall, the study underscores the promising potential of NLP methods in accelerating the pace of energetics research and facilitating the development of new energetic materials, while also highlighting areas for future exploration and improvement. [6]

2.1 Technologies for information extraction

A range of Natural Language Processing (NLP) techniques has been explored for knowledge discovery and information extraction within the field of energetics. These techniques include Latent Dirichlet Allocation (LDA), Word2Vec (W2V), and Transformer models. LDA, a probabilistic topic modeling technique, has been utilized to identify thematic patterns within large corpora of energetic documents, although its interpretability can be improved. Word2Vec, on the other hand, has demonstrated effectiveness in clustering thematically similar words, thereby aligning well with energetic motifs. Transformer models, such as BERT, have shown promise in capturing context and predicting masked words within energetics documents when fine-tuned on domain-specific data. However, challenges such as interpretability and computational constraints in hyperparameter tuning have been noted. Despite these limitations, these NLP techniques exhibit strengths in capturing critical concepts within energetics, highlighting their potential in accelerating research and facilitating the development of new energetic materials. Further exploration and refinement of these techniques are essential to harness their full capabilities and address existing challenges in the field.

A research gap emerges concerning the specific domain of information extraction from email content within the field of energetics. While existing studies have explored various NLP models for tasks such as sentiment analysis, summarization, and relation extraction, none have specifically addressed the challenges and nuances of extracting transactional information from emails within the energetics domain using BERT. This gap represents an opportunity for the current project to contribute novel insights and methodologies tailored to the unique requirements of extracting transactional data, such as purchase orders and invoices, from email communications within the energetics industry using BERT. Furthermore, while some research has touched upon email content analysis, the inclusion of attachments as a source of information extraction remains largely unexplored. By acknowledging that valuable transactional data can often be found within attachments, such as DOCX files, PDFs, or spreadsheets, the project aims to extend its scope beyond email bodies alone. Leveraging the capabilities of BERT and fine-tuning it for this

specific domain, the project aims to fill this research gap by developing a comprehensive framework capable of extracting transactional information not only from email bodies but also from attachments, thus advancing the state-of-the-art in information extraction from email content within the field of energetics.

NER has been found useful in the detection and classification of certain key entities related to energetics, such as chemical compounds, processes, and equipment mentioned in the energetic documents. Tagging such entities enables the creation of structured databases for advanced data mining and retrieval. Dependency Parsing of sentences oriented to grammatical structure will help to extract relationships among entities and enhance an understanding of complex descriptions and procedural texts typical in Energetics. Besides, techniques like transfer learning have allowed pre-trained models, which after adaptation with domain-specific tasks require relatively low data, maximizing the efficiency and remaining cost-effective. Hereby, though promising, such approaches are still challenged by the availability of high-quality annotated data and intrinsic complexity in the integration of many NLP techniques in the pipeline. Addressing these challenges philosophically requires further refinement and innovation, especially with respect to the customization of NLP frameworks to meet the specifics of the energetics domain. Continuous development in this field will lead to unlocking deeper insights in driving the advancement of research and practical applications in the industry.

3 METHODOLOGY

3.1 Theoretical Formulations

3.1.1 Extracting Email Data

Extraction of the content of the email would require parsing of the raw email data into parts, such as headers, the body, and attachments. This is in accordance with certain Internet standards, for example, RFC 5322. Normally, specialized email parsing libraries or modules would be used to perform this step since they would be written to understand the format of the email data. The subject and body of an email can then be extracted after parsing. The subject normally goes into a header field specifically reserved for this, and the main content of the message goes into the body. When the email is formatted with MIME—Multipurpose Internet Mail Extensions—the body may have various parts. Such parts allow attachments of different content types, such as plain text, images, or any other kind of attachment.

Many e-mail messages contain attachments, files attached to the e-mail message. In order to be extracted, the attachments first need to be identified within the structure of email data itself. This is usually by specific headers or content types. Afterwards, identified attachments will be extracted and saved as separate files. Now, if the need is to extract only .docx attachments, which are Microsoft Word documents, the extracted attachments will further have to be filtered according to their file extensions. This can be done by reading the name of every attachment or by its content type and saving only attachments with .docx extensions. Extraction of the subject, body, and all attachments in .docx form is done, and thereafter, the content can be further dealt with as desired in the application. For example, one can extract the text content from email bodies and from .docx attachments, after which, it could be handled by techniques of natural language processing or attachments could be converted to some other format for further processing.

3.1.2 Preprocessing

This is a multi-step process for the preprocessing of email content and attachments: to make a BERT model ready for fine-tuning efficiently with text data and extract information accurately. It starts with extracting the text content from the email body and its .docx attachments. In the case of email bodies, libraries like 'email' or 'BeautifulSoup'

can be utilized to parse and extract text, while 'python-docx' can deal with .docx attachments.The next critical step after extracting the text would be normalization. It needs converting all the text into lowercase for uniformity, removal of accents from the characters like making 'é' as 'e', expanding contraction like "don't" as "do not", common abbreviations, etc. Information that is irrelevant, including email headers and signatures, and any HTML or other formatting tags are removed. 'BeautifulSoup' and similar libraries may be used to strip the HTML tags. This step is important for the accuracy of subsequent processing and model training, as it ensures that only relevant content remains.The text, after normalization, should be properly encoded for various characters and languages if it is to be compatible with the BERT tokenizer. After cleaning, the text will then be fed into BERT's tokenizer to be tokenized. This means the text has to be broken down into individual words, sub-words, or word pieces processable by the BERT model. This basic stage of tokenization is quite important in the text preparation understandable and processable by BERT.

Variable lengths of email content are handled by applying sequence handling techniques. This involves truncating the text in excess of BERT's maximum input sequence length or splitting longer texts into multiple sequences. After that, padding will have to be done on every sequence to ensure that all input sequences are of uniform length, as required by BERT. Masking is applied to indicate which parts of the sequence are actual content and which are padding so that it can be ignored by the model.For a named entity recognition operation, entity annotations will have to be added to the text. Tags should be placed around relevant entities—such as names, dates, and locations—in the text. This will train the BERT model to identify and extract those entities during inference.If a labeled dataset is available, it becomes possible to split the preprocessed data into training, validation, and test sets. A training set will be used for fine-tuning the BERT model on the concrete task, a validation set for tuning hyperparameters and monitoring the model's performance during training, and a test set for the evaluation of the final model's performance. Such division will help to ensure that the model is well-trained and its performance can be accurately assessed.

3.1.3 Tokenization and Encoding

Tokenization and encoding are crucial steps in preparing text data for training a BERT model. This process transforms raw text into a form that the model can effectively process. It involves breaking down the text into smaller units called tokens, which can be words, subwords, or even characters.

BERT often employs WordPiece tokenization, which efficiently handles large vocabularies by splitting words into subwords and characters. For example, the word ”playing” would be tokenized as ”play” and ”##ing,” where ”##” indicates a subword continuation. This allows BERT to understand and generate novel words by combining learned subwords.

There are special tokens like ‘[CLS]‘ (classification token) and ‘[SEP]‘ (separator token) that provide context: ‘[CLS]‘ represents the entire sequence, while ‘[SEP]‘ marks the end of sequences in tasks involving multiple sentences.

The tokenizer can deal with unknown words by decomposing them into known subwords or characters, ensuring that even out-of-vocabulary or misspelled words can be handled.

Encoding then converts these tokens into numerical representations. Each token is mapped to a unique ID using a pre-defined vocabulary. For instance, ”play” might have an ID of 5831, and ”##ing” an ID of 2075.

The attention mask is a binary vector where 1 represents real content and 0 represents padding. This helps the model focus on relevant tokens during training and inference while ignoring padded elements.

There are special tokens like ‘[CLS]‘ (classification token) and ‘[SEP]‘ (separator token) that provide context: ‘[CLS]‘ represents the entire sequence, while ‘[SEP]‘ marks the end of sequences in tasks involving multiple sentences.

The tokenizer can deal with unknown words by decomposing them into known subwords or characters, ensuring that even out-of-vocabulary or misspelled words can be handled.

Encoding then converts these tokens into numerical representations. Each token is mapped to a unique ID using a pre-defined vocabulary. For instance, ”play” might have

an ID of 5831, and ”##ing” an ID of 2075.

The attention mask is a binary vector where 1 represents real content and 0 represents padding. This helps the model focus on relevant tokens during training and inference while ignoring padded elements. Segment IDs (optional) differentiate tokens from different sentences in tasks involving sentence pairs. These are typically assigned as 0 for the first sentence and 1 for the second.

Padding allows all input sequences to have the same fixed length required by BERT. Shorter sequences are padded with a special token, usually ‘[PAD]‘.

3.1.4 BERT Model Architecture

The BERT model architecture, having a transformer-based structure, is very effective in capturing context-based relationships among text words. In addition to this, it includes multiple layers of bidirectional transformers comprising attention heads and feed-forward neural networks pre-trained on large text corpora through masked language modeling and next sentence prediction tasks. This pre-training allows BERT to capture deep contextual relationships and semantic meanings in the text. For NER, input to the BERT model involves tokenized text sequences with special tokens like ’[CLS]’ at the beginning and ’[SEP]’ at the end. These tokens are added to help in sequence representation. Every word in the input sequence will be represented by a combination of token, segment, and position embeddings to retain the word order.

For example, the transformer layers of BERT are made up of multi-head self-attention mechanisms computing attention scores between all tokens so that each token attends to all other tokens in the sequence, capturing their contextual relationship. Feed-forward neural networks are applied to each token independently; on top are applied the activation functions, like GELU (Gaussian Error Linear Unit), with normalization and residual connections added in every layer for stability. For adapting BERT to NER, upon the output of BERT, a classification layer with softmax activation is added to predict the type of entities for every token. The model applies a token-level cross-entropy loss function to compare the predicted entity classes against the true label and optimizes this loss function by backpropagation and optimization algorithm AdamW.

The self-attention mechanism in BERT is particularly useful for NER, as it makes the model understand dependencies of words independent of their distance in the text and duly understand the context in which a named entity appears. For example, ”Apple is looking at buying a UK startup”; because of the context, the word ”Apple” will be correctly classified as an organization and not a fruit. The final output of the BERT model for NER would be a sequence of entity labels corresponding to each token, identifying named entities such as person, organization, and location. That is, it does accurate NER using the deep contextual understanding offered by BERT and powered by a very powerful transformer architecture, empowering it quite effectively in various applications of NER. Fine-tuning BERT further on domain-specific NER datasets makes the model very accurate in identifying and classifying the named entities within a text.

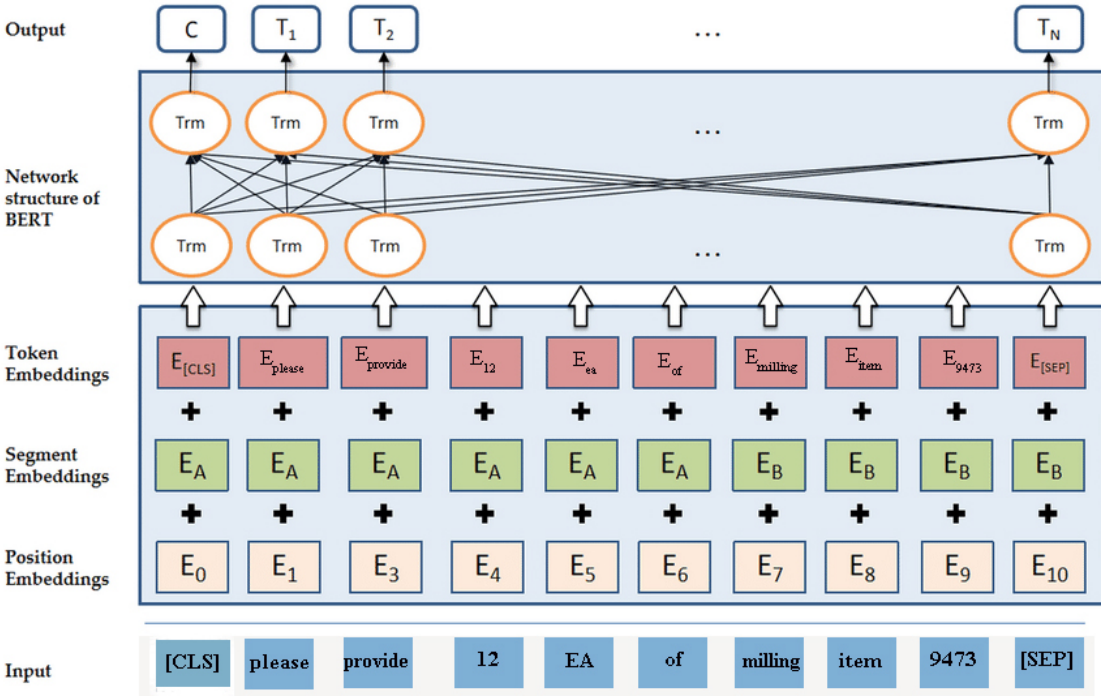

Figure 3.1: Overall Structure of BERT

3.1.5 Fine-tuning BERT for Information Extraction

It is especially applied to named entity recognition, relation extraction, and question answering. Fine-tuning is used to tune the pre-trained general language understanding capabilities of the BERT model on special tasks to enhance its ability to extract relevant information from text. The first step then would be to create a labeled dataset for adaptation trainable in this information extraction task, keeping in mind that annotated text can be for NER. It preprocesses text data by tokenizing it with BERT's tokenizer for subword splitting and adding special tokens like [CLS] at the beginning and [SEP] at the end. Specifically, this step converts tokens into IDs, prepares attention masks to tell apart real tokens from paddings, and generates segment IDs if necessary. Next, fine-tuning of the pretrained BERT model by adding a task-specific layer on top of BERT's last hidden states will be done. In this case, for NER, it will have to be a token classification layer that will assign entity labels for each token.

In this process of training, an appropriate loss function is used, such as token-level cross-entropy loss for NER, comparing predicted entity labels against the true labels. In this newly added task-specific layer and the whole BERT model, parameters are optimized with an optimization algorithm, with hyperparameters like the learning rate,

batch size, and the number of training epochs adjusted. Performance will be monitored to prevent overfitting using a validation set. Metrics for evaluating the model are precision, recall, and F1-scores. The final model is then tested, after training, to store how good it generalizes to new, unseen data.

3.1.6 Inference and Prediction

Inference and prediction with fine-tuned information extraction models will be run over new, unseen text data to predict such meaningful pieces of information like named entities, relationships, or answers. This will start with the basic step of preparing input text by tokenizing it using the BERT tokenizer, which involves breaking down a word into subword units that can fall within the range of its capacity, adding special tokens like [CLS] and [SEP]. These are then converted to token IDs, along with attention masks indicating which ones constitute the real content and which are padding. If necessary, segment IDs will be created, which aid in differentiating segments of text related to, for instance, multiple-sentence tasks.

After preparing the input, it is fed into the fine-tuned BERT model. The transformer layers of the model capture contextualized representations for all tokens in a sentence as an input to it, based on the surrounding context of each token. For NER and similar tasks, the output from BERT's final layer is fed through a token classification layer, where each token is assigned probabilities with regard to its likelihood of belonging to the different predefined entity types, such as a person or an organization.At prediction time, the model predicts the most likely entity label for each token in an input text considering these probabilities. These predictions are thus mapped back onto the initial tokens to come up with final annotated output in which every token is labeled according to its predicted entity type. This annotated output gives a structured representation of the extracted information from the input text.

3.1.7 Post-processing and Result

In the setting of fine-tuned BERT-based information extraction, by post-processing it means refactoring the model output to enhance both accuracy and usability. These could include normalization of entities to put them in standard form or shape of representation of an entity; linking entities against external databases in entity linking; in relationship

extraction, enthusiast approaches try to claim the links between entities; contextual enrichment gives more information around the entities; and formatting results clearly for presentability. Clearly, these steps guarantee that the extracted information is accurate and coherent from the input text data, so it can further be relied on during its integration to downstream applications for analysis; thus, maximizing utility in real-world tasks that require automated information extraction and analysis with a BERT model.

3.2 Mathematical Modelling

A language model forms the backbone of these applications. A language model is just a probability distribution. Given a sequence of words $w_{1:(t-1)} = (w_1, \ldots, w_{t-1})$, a language model gives the probability of all the words in your vocabulary V to follow this sequence,

$$P(w_t \mid w_{1:(t-1)}), \quad w_1, \ldots, w_{t-1}, w_t \in V. \tag{3.1}$$

With such a language model, one can generate new texts: Start with a sentence, then choose the word with the highest probability (or sample according to probabilities) and feed the new appended sequence back into the model to generate the next word. The language model can be used to assign a probability to a sentence (using the chain rule of conditional probabilities) as

$$P(w_{1:n}) = \prod_{i=1}^{n} P(w_i \mid w_{1:(i-1)}). \tag{3.2}$$

There are various methods to develop a language model. One approach involves attempting to encode all grammatical rules and word meanings into a computer program, but this is exceptionally challenging. Another approach, which has gained significant traction in recent times and has led to the creation of highly impressive language models, does not necessitate the explicit encoding of grammar or worldly knowledge. Instead, neural networks are trained using extensive amounts of text data, enabling them to learn how to construct coherent sentences solely from the information they are exposed to.

Transformer-powered models like BERT, which are also based on the encoder-decoder architecture, are bidirectional because they predict words based on the previous words and the following words. This is achieved through the self-attention mechanism, a layer

that is incorporated in both the encoder and the decoder. The goal of the attention layer is to capture the contextual relationships existing between different words in the input sentence.

Nowadays, there are many versions of pre-trained BERT, but in the original paper, Google trained two versions of BERT: BERTbase and BERTlarge with different neural architectures. In essence, BERTbase was developed with 12 transformer layers, 12 attention layers, and 110 million parameters, while BERTlarge used 24 transformer layers, 16 attention layers, and 340 million parameters. As expected, BERTlarge outperformed its smaller brother in accuracy tests.

$$\text{BERT}_{\text{BASE}}:$$

$$L = 12 \quad \text{(Number of layers, i.e., the total number of encoders)}$$

$$H = 768 \quad \text{(Hidden size)}$$

$$A = 12 \quad \text{(Number of self-attention heads)}$$

$$\text{Total Parameters} = 110\text{M}$$

$$\text{BERT}_{\text{LARGE}}:$$

$$L = 24 \quad \text{(Number of layers, i.e., the total number of encoders)}$$

$$H = 1024 \quad \text{(Hidden size)}$$

$$A = 16 \quad \text{(Number of self-attention heads)}$$

$$\text{Total Parameters} = 340\text{M}$$

3.2.1 Self-attention

Self-attention is a mechanism used in machine learning, particularly in natural language processing (NLP) and computer vision tasks, to capture dependencies and relationships within input sequences. It allows the model to identify and weigh the importance of different parts of the input sequence by attending to itself.Self-attention operates by

transforming the input sequence into three vectors: query, key, and value. These vectors are obtained through linear transformations of the input. The attention mechanism calculates a weighted sum of the values based on the similarity between the query and key vectors. The resulting weighted sum, along with the original input, is then passed through a feed-forward neural network to produce the final output. This process allows the model to focus on relevant information and capture long-range dependencies.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.3}$$

Q : Query matrix

K : Key matrix

V : Value matrix

d_k : Dimension of the key vectors

3.2.2 Fine tuning BERT

Mathematical equation that demonstrates the loss function used in training a model like BERT for named entity recognition (NER). This example uses the Cross-Entropy Loss, which is commonly employed for classification tasks:

$$\mathscr{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log\left(\hat{y}_{ij}\right) \tag{3.4}$$

$\mathscr{L}$: Loss function

N : Number of tokens

C : Number of classes

y_{ij} : True label indicator for class j and token i

$\hat{y}_{ij}$: Predicted probability for class j and token i

(x_i, y_i) : Input-output pairs in the training data

3.3 System Block Diagram

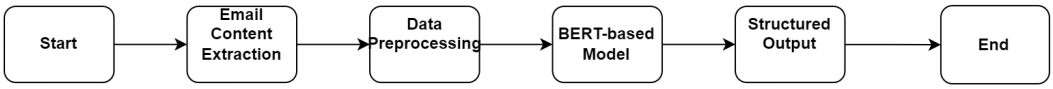

Figure 3.2: System Block Diagram

3.3.1 Email Content Extraction:

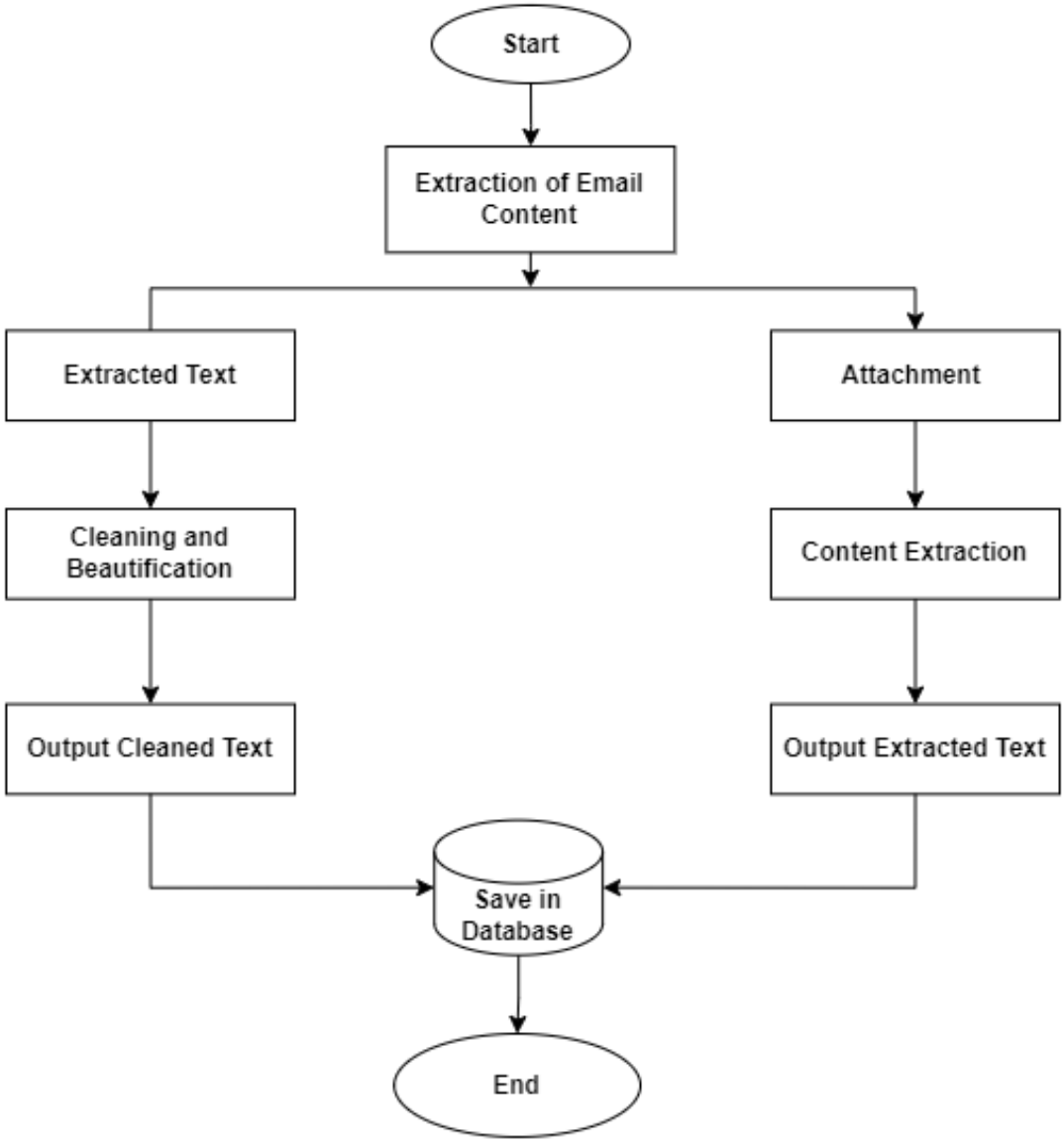

Figure 3.3: Flow Of Extracting Email Data

Email content extraction is a structured process of parsing the raw email data based on

Internet standards, particularly RFC 5322, which describes the format of email messages. This parsing shall become very instrumental in benchmarking key components like headers, body content, and attachments. Specific email parsing libraries are used in extracting the subject and body efficiently from the email in designated header fields. Attachments are also handled with precision since this is a very common feature within emails. The extraction process identifies attachments, especially those in ".docx" format, which are Microsoft Word documents, particularly relevant for their textual content. A filtering mechanism based on file extensions ensured that only ".docx" attachments would be kept for further processing.

Extracted and cleaned email content, structured data is then systematically stored in a database for processing and analysis. A few key components of email data have been considered in database design. The field content of the email, containing the main text together with the embedded information, is saved for easy retrieval for examination. For emails with attachments, in particular, in ".docx" format, the text from these attachments may be stored in a separate field for full-text analysis. Other important information about the sender, such as his email address and name, are also recorded for sorting purposes, in order to track the origin of the e-mails. The name of each attachment and related metadata are stored for later identification and access to specific attachments. In other words, an orderly approach like this in designing all the relevant components of the email ensures efficient querying, retrieval, and management of that data for scalable and effective handling of large volumes of email information.

3.3.2 Data Preprocessing:

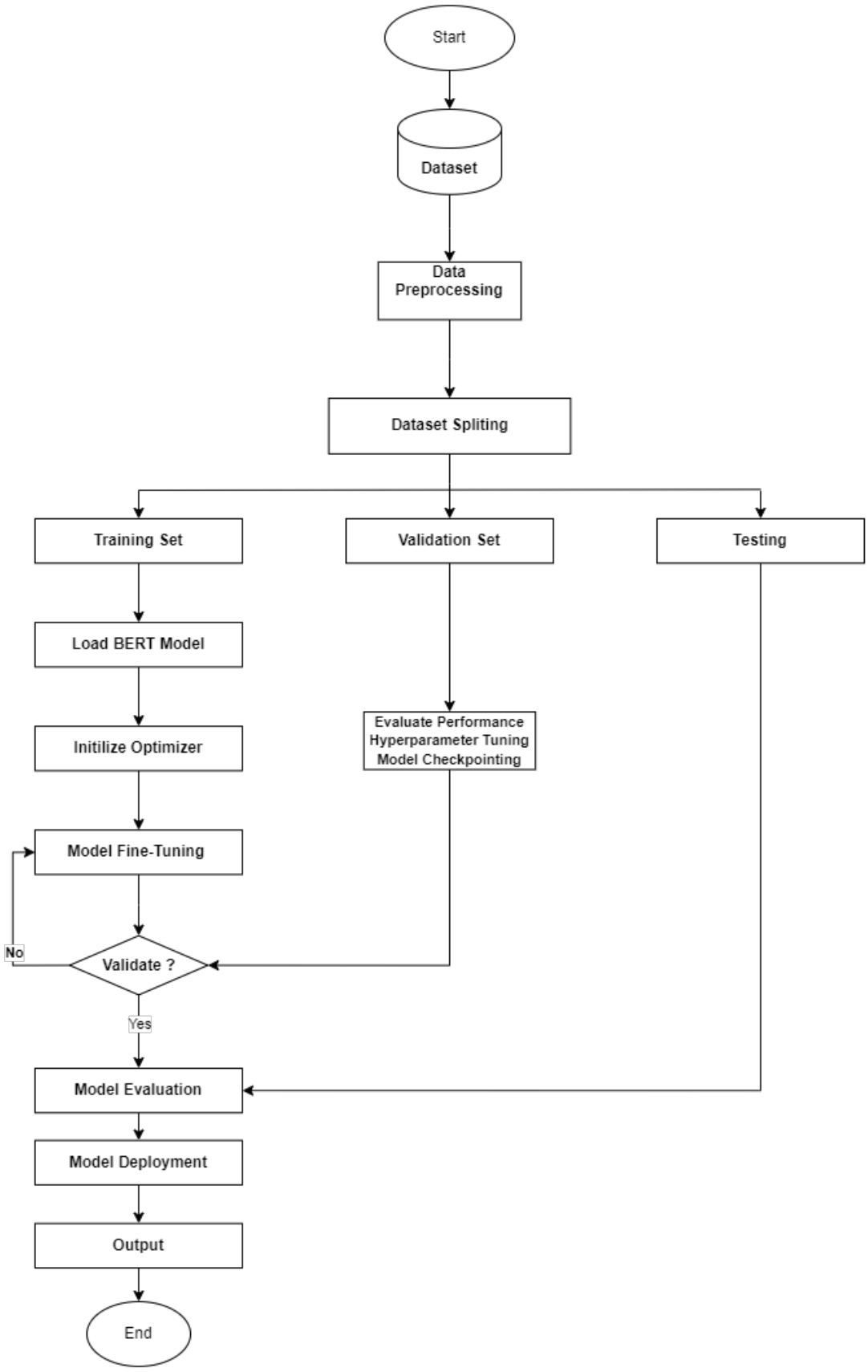

Figure 3.4: Flowchart for training BERT Model

In data preprocessing phase, it is essential to prepare all data extracted from the emails to ensure it is in the correct format for the model.The raw text goes through several critical steps in the data preprocessing phase for email analysis using BERT. Sentences are tokenized while keeping label alignment; special tokens ”[CLS]” and ”[SEP]” are added at the start and end, respectively. Then, the input is adjusted to a maximum length that is specified through truncation or padding, with labels modified accordingly. An attention mask is then created that differentiates between real content and padding, letting the model zero in on information with meaning. This thorough preprocessing guarantees that email data will be optimally formatted for BERT to work its magic and conduct effective and informative analysis while ensuring extracted information integrity.

3.3.3 BERT Model:

The ”BERT-based Model” block represents the core component of the system, where the preprocessed email content is passed through a fine-tuned BERT model. BERT, which stands for Bidirectional Encoder Representations from Transformers, is a deep learning model developed by Google that has demonstrated remarkable performance on various natural language processing tasks. By leveraging BERT, the system generates contextualized word embeddings that capture the semantic meaning of the email content, taking into account the relationships between words in the text. These embeddings serve as rich representations of the email content, providing a foundation for subsequent stages of analysis and extraction.

Model Initialization:

For the training BERT model first step is to initialize pre-trained BERT base model for general language understanding, which has been trained on an extremely large corpus of unlabeled text. This is then fine-tuned on our specific labeled dataset, where its pre-trained weights are tuned for the classification task at hand. In this case, since BERT uses a transformer-based architecture with self-attention mechanisms, it is versatile with respect to all kinds of NLP tasks. We also define the optimizer that couples adaptive learning rates with weight decay regularization. A learning rate scheduler helps find a better optimum by dynamically changing the learning rate in the course of training.

Gradient clipping helps to avoid exploding gradients and hence makes the training more stable.

Training Model:

Training of model is done through training process which is structured around epoches , where each epoch represents a complete pass through the entire training dataset. This repetition allows the model to see the data multiple times, gradually refining its understanding and improving its performance.The training process for each epoch processes data in batches. Each batch will pass through a forward pass, where the model ingests the tokenized text inputs and corresponding attention masks, makes some kind of prediction, and calculates the loss against the true labels. This loss, as a measure of the accuracy of predictions, is accumulated over batches to track performance on the epoch level. Backpropagation then computes gradients of the loss with respect to model parameters, which an optimizer would use in updating these parameters to eventually decrease future loss. Gradient clipping avoids instabilities due to huge gradients. Accuracy is also calculated by comparing concrete predictions derived from the output logits with the true labels to obtain an intuitive metric of performance. In this way, with the loss and accuracy per epoch, we have a way of monitoring how the model learns and noticing problems such as overfitting and underfitting, and recognizing when training must be stopped. This iterative process of prediction, computation of error, and adjustment of parameters is at the heart of the learning loop and continues until a predetermined epoch count or other stopping criterion is reached.

Validating Model:

The model's performance is evaluated on the validation set , validation are crucial for monitoring overfitting and adjusting hyperparameters .with the combination of with early stopping mechanism it ensure no overfitting in the optimization of model performance .After every training epoch, the model is tested on a separately held-out validation dataset. Basically, this computes loss and accuracy without touching model parameters to give an unbiased performance estimate on unseen data. Validation metrics provide diagnostic

values for overfitting and guide hyperparameter tuning, early stopping decisions, etc. In the case of early stopping, it stops training when validation performance has not improved for a specified number of epochs; hence, overfitting will not occur, and the model that is best in generalization performance will be chosen. This combined approach will hopefully give the proper balance between train data performance and generalization toward new data—something very critical in real-world applications.

Testing Model:

Final evaluation of the model is done on the test set which is very important in obtaining the actual performance and generalization of the model. It processes test data in batches and makes predictions without any update to the parameters. Computed test loss and accuracy provide the final word about the model's ability to generalize to new examples. This is what will be used to measure the performance of the model in real-world applications. Comparing test results with training and validation performance identifies the potential overfitting or underfitting of the model and helps to decide further refinement. This is a very critical evaluation step to prove the effectiveness and reliability of the model in practical applications.

3.3.4 Output:

Output from the model is generated by passing input data to the model. This model's output is obtained by passing input data through a forward pass inside a torch.no_grad() context to ensure that no gradient calculation is done during inference. Then, calling model(**inputs) returns an output with logits, which corresponds to raw predictions for each token against all possible entity classes. These logits are drawn from the attribute outputs.logits, while the predicted labels are derived by applying torch.argmax() to the logits along the second dimension, selecting the most likely entity class for each token. This process transforms the model's high-dimensional output to a sequence of numerical label indices which are then mapped to human-readable entity labels using a predefined label map. The predicted entity labels are then aligned with their corresponding input tokens, excluding special tokens, and joined to form a coherent sequence of token–entity pairs representing the results for entity recognition on the given input text.

3.4 Instrumentation Requirements

The instrumentation requirements for information extrction form email project involve the tools, software, and hardware needed to develop, test, and deploy the system effectively. Here's an overview of the instrumentation requirements:

Hardware Devices: To support tasks such as data preprocessing, training sophisticated language models like BERT, and extracting information from text, a combination of local and cloud-based computing resources is utilized. Local resources include a Windows operating system with 16 GB RAM and a 500 GB SSD, sufficient for development and smaller-scale tasks. For more intensive tasks, Google Colab GPU instances are leveraged, providing access to high-performance computing capabilities necessary for model training and inference.

Software Requirements:

1. **Python Environment**:
    - Python serves as the primary programming language for building the information extraction pipeline. The Anaconda distribution with Python is used, along with Jupyter Notebooks for interactive development. Key libraries such as PyTorch, and Hugging Face's Transformers are essential for integrating BERT models into the pipeline.Django is used to develop the graphical user interface (GUI), providing a robust framework for frontend development and interaction with the information extraction system. Celery is employed for running background tasks, such as fetching new emails from mailboxes asynchronously, ensuring real-time updates and responsiveness.
2. **BERT Models**:
    - Pre-trained BERT models are essential for conducting language comprehension tasks, including tokenization, generating contextual embeddings, and extracting information. We plan to utilize BERT-base, BERT-large, or

specialized BERT models fine-tuned for particular domains. Accessing these pre-trained models can be achieved by downloading them from the Hugging Face Model Hub or by training custom models internally using the resources available to us.

3. **Development Tools**:

   - Development tools are indispensable for coding, debugging, and maintaining version control of the projects codebase. We plan to utilize integrated development environment / IDE as Visual Studio Code, along with Git for version control. These development tools can be readily installed on local machines.

4. **Data Management tools**:

   - Data management tools are essential for organizing, retrieving, and manipulating email datasets and extracted information efficiently. We are considering the use of relational databases such as MySQL databases , for storing and accessing data. These tools can be installed on local machines.These tools facilitate efficient data storage, retrieval, and management, supporting the scalability and performance requirements of the project.

3.5 Dataset Explanation

The dataset is very important for this project, as it forms the base through which the BERT model will be trained to do entity recognition on transactional data extracted from emails. The alignment of the dataset relevance is further underpinned by its direct definition towards meaningful information extracted from emails on transactions, including details of a transaction, sender information, purchase orders, item numbers, and quantities. The quality and structure of this data influence the performance of the system in correctly identifying and classifying the entities. An organization provided data for the dataset, which is real business-oriented data with minimal name manipulations to protect its consumers. Each instance in the dataset typically has these elements. Therefore, the model is being exposed to relevant and representative samples of the kind of information it will be required to extract.

Manual pre-processing of the raw email text to prepare transcripts for training the model. It began with the extraction of relevant parts of the email body, especially those with information on transactions. Each of those had careful annotation to identify the entities, which is the forming of clear examples of what is to be identified and classified by the model. Much attention to the cleaning of the data was specifically directed at removing the amount of extraneous information and normalizing any differences in format. Finally, cleaned and annotated text was put into structured format which the BERT model can actually take in as input. Such normalizations comprise converting any date into a standard format, ensuring numerical quantities are represented in a consistent way, and normalizing sender and receiver details. The provision of high-quality, well-annotated, and normalized data will put a good setting for the model in learning to generalize by drawing inference from the training examples and correctly identifying the entities present in a new unseen email. The careful preparation of the data enhances the model's ability to perform reliably on deployed systems, ensuring that it can extract meaningful and actionable information from transactional emails with high precision.

Each data point in the dataset typically includes the following components:

Table 3.1: Dataset Sample

Body	**Po_num**	**Item_des**	**Qty**	**Unit**	**Cost**	**Sender**
Hi msc, This is to confirm our order 4241463560 for 18 EA of Supplies (Item ID: DX-ESX20PATH). The total is $23. Best regards, SSAutomotive	4241463560	DXESX20PATH	18	EA	23	SSAutomotive

The data were collected from various business transaction emails. While gathering the

dataset, there were some missing values in the collected dataset. Before proceeding with the dataset, we need to preprocess it. Handling missing values in a dataset depends on the nature of the data and the importance of each column.In order to handle missing value in body column , we have drop the row that containing empty body value because email text is necessary for Named Entity Recognition tasks, and missing values are unlikely to provide useful information. For the remaining data , if the value is still missing for some column and the number of missing value row is very low then manually add missing value by evaluating the body content . In some case if the value is not present then those in the empty place UNKNOWN key are used in the dataset .

	subject	body	sender	receiver	order_number	Item_ID	Product_Name	Cost	Unit_Of_Measure	quantity
1	Order #4064514301 Received	Greetings USPS, There seems to be an issue wit...	robertPharmaceutical	USPS	4064514301	11570447	Wipers/Rags	85.55	EA	8
6	New Product Recommendation - Order 4571071293	Hi UPS, We noticed a new variant of the Suppli...	owensandminnor	UPS	4571071293	63595813	Supplies	4.05	EA	10
8	Order Confirmation - 4235058282	Dear hospeco, We are ordering (4235058282) 14 ...	concordanceHealth	hospeco	4235058282	00919423	Trimmers	277.57	EA	14
9	New Product Recommendation - Order 4770340052	Hello USPS, We are placing an order (477034005...	SSAutomotive	USPS	4770340052	14145445	Trimmers	357.97	EA	11
13	Order 4009702962 Out of Stock	Dear essendant, Thank you for offering free sh...	owensandminnor	essendant	4009702962	03884301	Ventilation	862.36	EA	18
...	...	...	...	...	...	...	...	...	...	...

Figure 3.5: Snapsot of Dataset

This is a snapshot of the dataset to provide a visual representation of the data we will be working with. For training the BERT model for information extraction, we have utilize a comprehensive dataset consisting of over 20,000 entries. This dataset is diverse and robust, ensuring that the model will be well-trained and capable of handling various information extraction tasks effectively.A significant portion of this dataset has been collected from attachments in emails. This means we have curated real-world examples where important information is embedded within attached documents, reflecting practical scenarios where the BERT model will be applied. By incorporating data from email attachments, we ensure that the model is exposed to various document types and formats, enhancing its ability to generalize and perform accurately in real-life applications.

We cleaned up the dataset, then tokenized the body field sentences into tokens for the

BERT model's NER approach, labeling these tokens based on their presence in several columns. Tokens at each position matching the values in these columns were labeled accordingly; otherwise, when not found in any column, they were labeled as 'O'. These labeled tokens were then stored in a new column called 'WordLabel'. At the end of this process, our dataset would be ready with the sentences tagged appropriately for NER to train the BERT model.

Example of labeled token:


```
{"Hi": "O"},
{"msc": "O"},
{",": "O"},
{"This": "O"},
{"is": "O"},
{"to": "O"},
{"confirm": "O"},
{"our": "O"},
{"order": "O"},
{"4241463560": "B-ORDER_NUMBER"},
{"for": "O"},
{"18": "B-QUANTITY"},
{"EA": "B-UNIT"},
{"of": "O"},
{"Supplies": "O"},
{"(": "O"},
{"Item": "O"},
{"ID": "O"},
{":": "O"},
{"DX-ESX20PATH": "B-ITEM_ID"},
{")": "O"},
{".": "O"},
{"The": "O"},
{"total": "O"},
```

```
{"is": "O"},
{"$": "O"},
{"23": "B-COST"},
{".": "O"},
{"Best": "O"},
{"regards": "O"},
{",": "O"},
{"SSAutomotive": "B-SENDER"},
```


3.6 Description of Algorithms

Algorithm Used in BERT

BERT relies on a specialized deep learning architecture known as a Transformer. In this architecture, the Transformer Encoder serves as the fundamental component of BERT. BERT is different than traditional recurrent neural networks (RNNs) that process information sequentially, the Transformer can analyze all parts of a sentence simultaneously. This capability enables BERT to capture long-range dependencies between words and comprehend the context of a word based on its surrounding words in the sentence.

At the heart of the Transformer Encoder lies the Self-Attention Mechanism, which allows the model to focus on specific segments of the input sentence relevant to understanding a particular word. By attending to different parts of the sentence dynamically, the model grasps the intricate relationships between words across the entire sentence.

BERT employs two key pre-training techniques to enhance its language understanding capabilities. The first technique, Masked Language Modeling (MLM), involves masking some words in a sentence and tasking the model with predicting these masked words based on contextual cues. This process aids BERT in developing a profound understanding of word relationships within a sentence.

The second pre-training technique, Next Sentence Prediction (NSP), involves presenting the model with pairs of sentences and prompting it to predict whether the second sentence logically follows the first one. By mastering the relationships between sentences and their flow, BERT gains insights into constructing coherent text.

With the integration of these algorithms, BERT achieves a sophisticated understanding of language, empowering it to excel in various natural language processing (NLP) tasks. While the intricacies of these algorithms can be complex, this overview provides a foundational understanding of the core principles driving BERT's capabilities.

Algorithm Description for Extracting Email Content

Algorithm 1 Email Content Extraction

Require: *raw_email*: Raw email data

Ensure: *subject*, *body*: Extracted subject and body of the email, *docx_attachments*: Extracted ".docx" attachments

1: $parsed\_email \leftarrow$ parse_email(raw_email) ▷ Parse raw email data
2: $subject \leftarrow parsed\_email.subject$ ▷ Extract subject from parsed email
3: $body \leftarrow parsed\_email.body$ ▷ Extract body from parsed email
4: $attachments \leftarrow parsed\_email.attachments$ ▷ Extract attachments from parsed email
5: $docx\_attachments \leftarrow$ filter_attachments_by_extension($attachments$,"$.docx$") ▷ Filter attachments by extension
6: **return** $subject$, $body$, $docx\_attachments$

The given pseudocode encapsulates a comprehensive algorithm designed to extract essential information from raw email data. At its core, the algorithm starts by parsing the raw email data, a critical initial step that involves breaking down the email into its constituent parts such as headers, body, and attachments. Once parsed, the algorithm proceeds to extract the subject and body of the email, crucial elements that often encapsulate the primary content and context of the message. Furthermore, it identifies and retrieves any attachments included within the email, recognizing their potential significance in conveying additional information or resources. A pivotal aspect of the algorithm lies in its filtering mechanism, specifically tailored to sift through the attachments and selectively retain only those designated with a ".docx" extension, indicative of Microsoft Word documents. This strategic filtration ensures that the subsequent processing focuses solely on relevant attachments, streamlining the extraction process and enhancing efficiency.

Ultimately, the algorithm concludes by returning the extracted subject and body of the email, alongside the filtered subset of ".docx" attachments, thus furnishing users with refined, pertinent data ready for further analysis, manipulation, or presentation. Through its systematic approach and meticulous handling of email content, this algorithm serves as a valuable tool for extracting actionable insights and facilitating seamless integration of email data into diverse applications and workflows.

Algorithm 2 Filter Attachments by Extension

Require: *attachments*: List of attachments, *extension*: Desired file extension

Ensure: $filtered\_attachments$: List of attachments with the specified extension

1: $filtered\_attachments \leftarrow []$
2: **for** *attachment* **in** $attachments$ **do**
3: **if** attachment.extension == extension **then**
4: $filtered\_attachments$.append($attachment$)
5: **end if**
6: **end for**
7: **return** $filtered\_attachments$

The algorithm Filter Attachments by Extension is designed to filter a list of attachments based on a specified file extension. It accepts two inputs: attachments, a list of attachment objects, and extension, the desired file extension to filter by. The output is a list named filtered_attachments, which contains only those attachments that match the specified extension.The algorithm begins by initializing an empty list called filtered_attachments to store the filtered results. It then iterates over each attachment in the attachments list. During each iteration, the algorithm checks if the extension attribute of the current attachment matches the desired extension. If a match is found, the attachment is appended to the filtered_attachments list.After the loop has processed all attachments, the filtered_attachments list, now containing only the attachments with the specified extension, is returned as the result. This algorithm ensures efficient filtering of attachments based on file extension, making it useful in scenarios where specific types of files need to be isolated from a larger collection.

3.7 Elaboration of Working Principle

The working principle of the information extraction from email project revolves around leveraging advanced machine learning models, particularly transformer-based architectures like BERT, to understand and interpret the text. Here's an elaboration of the working principle:

3.7.1 Email Content Extraction:

In the Django application, there is a background task running to schedule, watching for and capturing newly arrived, unseen emails. This task fetches email metadata, such as the subject and details of the sender, with the full email body and attachments. In case of new mail detection, all attachments are downloaded and stored in the application environment under a specified directory. After the attachments have been stored, processing is done on ".docx" attachments. Each document is cleaned to allow only relevant text content to be extracted and preserved. It usually involves stripping off formatting artifacts, many other non-essential metadata, and return clean and structured text that might be fed into further analysis or other systems. The extracted content from these ".docx" files is then stored in a database. This database-centric approach facilitates efficient retrieval and management of extracted information, supporting applications that rely on structured data for tasks such as automated information extraction, content analysis, or document indexing.

When fetching unseen email data, the emails data are received in a raw format that includes a huge amount of encoded data. Our mail interest lies in extracting specific components of the email: the sender's name, sender's email, and the subject . While extracting these elements is easy as we are using RFC 5322 format for fetching but processing the email body requires additional steps due to the multipart structure of the document.

Here is a segment of the raw email data received and the cleaned data extracted from it:

Sample Segment of Raw Email Data:

```
MIME-Version: 1.0
From: Tika sah <tikasah84@gmail.com>
Date: Wed, 3 Jul 2024 21:48:02 +0545
Message-ID: <CAMUvmDQjJxccq6yJNooXQh82ULVXzFgH37XX_zZUD10Ux2BVhw@mail
.gmail.com>
Subject: Test order from docx
To: tika.sah@gsa-cs.com
Content-Type: multipart/mixed; boundary="00000000000083cba8061c59f5f2"

--00000000000083cba8061c59f5f2
Content-Type: multipart/alternative; boundary="00000000000083cba6061c59f5f0"

--00000000000083cba6061c59f5f0
Content-Type: text/plain; charset="UTF-8"

Greetings,

We are pleased to inform you that we have received a purchase order from RPG
Of Carolina , LLC. Attached to this email, you will find the purchase order
document in PDF format for your reference.We are requesting 15 pk of paper cup
with price 14.2 with order Purchase Order Number:
4517572516

--00000000000083cba6061c59f5f0
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"ltr">Greetings,<br><br>We are pleased to inform you
that we have received a purchase order from RPG Of Carolina , LLC.
Attached to this email, you will find the purchase order document in PDF
```

format for your reference.We are requesting 15 pk of paper
cup with price 14.2=C2=A0 with order Purchase
Order Number: 4517572516

--00000000000083cba6061c59f5f0--

Cleaned Email Data:

- **Sender Name:** Tika Sah
- **Sender Email:** tikasah84@gmail.com
- **Subject:** Test order from docx
- **Email Body:** Greetings,
  We are pleased to inform you that we have received a purchase order from RPG Of Carolina , LLC. Attached to this email, you will find the purchase order document in PDF format for your reference.We are requesting 15 pk of paper cup with price 14.2 with order Purchase Order Number: 4517572516
- **Attachments:** ['test_850_order_docx']
- **Attachments Content:**
  Greetings,
  We are pleased to inform you that we have received a purchase order from RCG Of North Carolina, LLC. Attached to this email, you will find the purchase order document in PDF format for your reference.
  We are requesting 15 pk of paper cups with a price of 14.2.
  Purchase Order Number: 4517572516
  Purchase Order Date: 07/03/2024
  Sender: RPG Of Carolina, LLC
  Thank you.

```
Best regards,
TEST Network
```

After these steps, we can ensure that the extracted email data is clean, structured, and ready for further processing or analysis. Additionally, for the attachments in the email, the content is extracted and saved in the database as attachment details. This method is essential for automated systems that process large volumes of emails and need to extract relevant information efficiently. Once we have the structured data from the emails, we save it in the database for further processing.

3.7.2 Preprocessing:

The preprocessing phase plays a crucial role in readying the email content for analysis by the BERT model.This will involve a series of transformations that prepare the raw text data for input to the model. First, the sentence has to be tokenized, and in this case, alignment with the respective word labels has to be preserved. Special tokens, namely ”[CLS]” and ”[SEP],” will be prefixed and postfixed onto the tokenized sentence, respectively, with ”O” (Outside) labels inserted for such tokens. Truncation and padding are applied to make the tokenized sentence and its labels of a certain maximum length. In case the tokenized sentence is longer than the maximum length, the method truncates it and its corresponding labels. In case it is shorter, [PAD] tokens will be added to achieve the desired length. The labels are also padded with ”O” to align them with the sentence after tokenization. An attention mask will be created. It will allow the BERT model to learn where actual content ends and padding tokens begin during training. The attention mask should be a binary list whereby each position is marked 1 if corresponding to a real token and 0 if corresponding to padding token. This mask lets the BERT model know that in its computations, it should take care not to focus on the padding but consider only meaningful tokens.

Input:

Sentence: Please provide 12 EA of miling item 9473....

After tokenization and adding special tokens: `[CLS], please, provide, 12, ea, of, mil, ##ing, item, 94, ##7, ##3, [SEP]`

Encoded IDS: `[ 101, 3531, 3073, 2260, 19413, 1997, 23689, 2075, 8875, 6365, 2581, 2509, 102]`

Attention Mask: `[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]`

Where 101 is the ID for [CLS], 102 for [SEP], 0 for [PAD], and the other numbers are token IDs.

After preprocessing of dataset, The entire dataset is divided into 3 porsion . This division has a common ratio of 70:20:10, which means that 70% of it is used for training, 20% for validation, and 10% for testing. The division serves specific purposes: it's the largest portion for teaching the model, another is the Validation Set for fine-tuning and preventing overfit, and the smallest portion is the Testing Set—kept aside for the final, unbiased evaluation of the model's performance. This strategic splitting will ensure that the model is learning well and can be optimized during development, and it can estimate correctly on unknown data to ultimately come up with a more robust and reliable machine learning model.

3.7.3 Model Generation:

Model training, specifically a BERT-based token classification model, comprises a few vital phases where the contributions in total impart capacity to learn for the model, hence leading to accurate predictions. First, pass the preprocessed and pre-tokenized data through the BERT model's embedding layer. This layer converts the token IDs to dense vectors known as embeddings. These embeddings capture the semantic meaning of the tokens by putting them as vectors in a high-dimensional space. This initial transformation is necessary since, if it wasn't there, textual data would not have meaningful numerical representations on which the subsequent layers can manipulate and learn.The output

from the embedding layer is then fed through multiple transformer layers of the BERT architecture. At the heart of the model lie these transformer layers in which self-attention mechanisms have been used to find the contextual relationship among tokens. Self-attention allows the model to weigh the importance of each token in a sequence relative to all other tokens. It can use this contextual understanding for such tasks as named entity recognition, where the meaning and label of the words are determined by its surrounding words.

In the process, positional encodings are summed with embeddings as data flows through transformer layers. These encodings enable the model to hold the sequence of the tokens, which is very important because the meaning of a sentence can totally turn around upon the order of words. Finally, outputs from these transformer layers will be a dense, context-informed representation of input tokens, ready for classification.In the final step, the information processes to a token classification layer. This layer applies a linear transformation to the outputs of the transformer layers to give logits for every token. Logits are unnormalized scores that indicate how likely each label is for a given token. The Cross-Entropy Loss function is applied to compute the difference between the logits predicted and the true labels. It is a measure of how much the predicted and true labels differ; the smaller the value, the better the performance of a model.In training, this loss will be minimized by the optimizer by modifying the weights of the model in accordance with gradients computed during backpropagation. First, it will compute the gradients of the loss function with respect to the model's parameters. Then, it updates the parameters in the direction that reduces loss using these gradients. This is to ensure that gradients do not become too large, enabling stable training. The clipping of gradients within a maximum value ('MAX_GRAD_NORM') allows learning to be stable and efficient. The following hyperparameters were used in the training process:

- **MAX_LEN**: 100
- **TRAIN_BATCH_SIZE**: 8
- **VALID_BATCH_SIZE**: 4
- **EPOCHS**: 5

- **LEARNING_RATE**: 2e-05
- **MAX_GRAD_NORM**: 10

Training occurs iteratively over some epochs; an epoch here denotes one complete pass through your training dataset. In each epoch, data is processed in mini-batches for efficient use of computational resources and incremental updates to the model's parameters. At the end of every batch, training loss is tracked to monitor how the model is learning, together with the training accuracy. The training accuracy will be computed by matching predicted labels against the true labels, considering active labels not set to '-100'.By the end of training, in the final step of training, it learned how to set its parameters such that the loss function was minimized, hence learning a function in a position to make accurate label predictions for tokens from their sentence contexts. Now properly trained, it's ready to generalize beyond the examples seen in training and make label predictions on new, previously unseen data.

3.7.4 Post Processing of Model Output:

After the BERT-based token classification model returns logits for every token, these final logits have to be transformed and refined to ultimately yield the last predicted label. It starts with applying argmax to the logits, returning the label with the highest probability per token. During this operation, the logits are converted into discrete predicted labels corresponding to the various entity classes defined in the task, like person, location, and organization.After generating the predicted labels in this approach, it becomes essential to filter out the labels assigned to subword tokens. In preprocessing, the label for a word was only maintained on the first token and other subword tokens are all labeled as '-100' to be ignored. Therefore, in the post-processing phase, these '-100' labels must be filtered out to make the result accurate. This will be ensured by considering active labels, that is, those which are not set to '-100', for comparison against true labels.The following is the computation of accuracy. Accuracy is computed by comparing predicted labels against true labels for each token. The labels will be flattened into one dimension, and the prediction for active tokens will simply be compared with their true labels. It uses the 'accuracy_score' function of the 'sklearn.metrics' library, which computes the proportion of correctly predicted labels out of the total count of active labels. This measure provides

a straightforward, intuitive metric to evaluate model performance.

Other evaluation metrics, such as precision, recall, and the F1-score, could also be computed to give an all-rounded assessment of how good the model is. Accuracy alone may not suffice in the case of imbalanced datasets. Precision answers the question of how many true positives the model has predicted compared to all positive predictions, while recall is a measure of the number of true positive predictions by the model out of actual positives in the dataset. The F1-score is just the harmonic mean of precision and recall, and thus it provides a single metric balancing the two aspects. when the model's performance has been evaluated, the predicted labels can be mapped back to their actual textual representations if needed. This is just the reverse process of the encoding done on the labels—that is, conversion of numerical label IDs into correspondent label names, like, e.g., '8' back to 'B-ITEM_ID' for a item entity. These labeled tokens can further be fed into a variety of applications dealing with named entity highlighting in the text, extraction of structured information from unstructured data, or integration of the extracted information into another system.In other words, this means logit conversion to predicted labels, filtering for subword token labels, computation of relevant evaluation metrics, and mapping numerical labels back to original textual form on the model output. Such a set of operations transforms the raw model output into actionable and interpretable results that will enable the practical application of the trained model in real-world scenarios. All these steps take good care of how accurate, meaningful, and ready for further use in a lot of NLP tasks the final output will be.

3.8 Verification and Validation Procedures

Verification and validation (V&V) are essential processes in assessing the accuracy, reliability, and effectiveness of a system or model. These processes ensure that the system meets its intended requirements and produces outputs that are trustworthy and useful.Verification involves ensuring the data used for training and testing is accurate and complete, the model is implemented correctly according to the chosen architecture and configuration, and the requirements are met. Validation involves ensuring the data is representative of real-world scenarios, evaluating the model's performance and generalization capabilities on held-out test sets or real-world data, and confirming the model meets user expectations and requirements through user studies and feedback

1. **Accuracy**: Accuracy measures the proportion of correctly classified instances out of the total instances evaluated. It's fundamental for assessing overall correctness and effectiveness. Accuracy is calculated as the ratio of correctly predicted instances to the total instances evaluated.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \tag{3.5}$$

2. **Precision**: Precision evaluates the proportion of true positive predictions among all positive predictions made. It's crucial when minimizing false positives is essential. Precision is calculated as the ratio of true positive predictions to the sum of true positive and false positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.6}$$

3. **Recall (Sensitivity)**: Recall measures the proportion of true positives correctly identified by the model out of all actual positives. It's particularly important when avoiding false negatives is critical. Recall is calculated as the ratio of true positive predictions to the sum of true positives and false negatives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.7}$$

4. **F1 Score**: F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is useful when there is an uneven class distribution.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.8}$$

The validation function valid computes these metrics by evaluating the model on the testing dataset. Here's a breakdown of how each metric is calculated and reported:

1. **Validation Loss:**The function computes the average validation loss by summing the loss values for each batch and dividing by the number of batches. This loss indicates how well the model is fitting the validation data.

2. **Validation Accuracy:**Accuracy is calculated by comparing the predicted labels with the true labels, excluding tokens labeled -100. The function computes the average accuracy over all evaluation steps.

3. **Precision, Recall, and F1-Score:**Using the seqeval library, the function generates a classification report that includes precision, recall, and F1-score for each class. This library is designed for sequence labeling tasks, making it suitable for NER evaluation.

4 RESULTS

The result of our project will be explored in this section . we will examine the results generated by our system and looking for possible errors in the predictions our model made. Our model has to find and classify several entities: sender, item ID, quantity, unit of measure, cost, and order number from the text relating to purchase orders. We reached a high accuracy during training, but some predictions were misclassified. We shall take one such case to illustrate the character of these errors and discuss some of the possible causes of misclassification.

We will explore two distinct aspects of our system. The first one will check if the system can successfully extract content from an email and a DOCX attachment. This will be a critical step to ensure that information within it gets appropriately extracted for future processing. The second one will extract information from the text content obtained in the previous step. It means that our trained model will identify and classify the key entities, ensuring data extracted from the extracted text is accurate and reliable. We intend to evaluate these two factors so that the general performance and effectiveness of our system in handling email content and attachments can be established.

4.1 Extracting Email Content:

The most important feature of this project is to get content from the UNSEEN email and proceed for information extraction . The process of extracting content from email is powered by feature of background task which run on scheduler basis and extracting content smoothly.The email content extraction process is an ongoing background task powered by Celery, fetching new and unseen emails from the inbox on a scheduled basis, so that incoming data is consistently getting scraped. Once the emails are fetched, it extracts text from the main body, headers, and any HTML content with remarkable granularity. This is subsequently followed by a cleansing operation that helps to get rid of noises like HTML tags, special characters, and formatting artifacts into a cleaned and homogenous piece of text that can be further processed.After cleaning the content of email , it is stored in database safely for further process.

Besides the email text processing,system have also capabilities to process attachments in the email and stored the content of attachment in different field . In the email if the attachments in DOCX are availabe then these attachments are processed. The script discovers and downloads these attachments and extracts their text content using the python-docx library. Then the text is cleaned and saved in the database. From the corpus database, the cleaned email content and extracted text are compiled into a format that shows known entities in a way that an end user can easily identify.

Raw email content while extracting:


```
[(b'1929 (FLAGS (\\Seen) RFC822 {20503}', b'Return-Path: <
    tikasah84@gmail.com>\r\nDelivered-To: tika.sah@gsa-cs.com\r\
    nReceived: (qmail 12398 invoked by uid 89); 31 Jul 2024
    17:15:24 -0000\r\nReceived: by simscan 1.4.0 ppid: 12394, pid:
    12396, t: 0.5059s\r\n scanners: attach: 1.4.0\r\nReceived: from
     unknown (HELO mail-wm1-f52.google.com) (209.85.128.52)\r\n by
    mail.gsa-cs.com with SMTP; 31 Jul 2024 17:15:24 -0000\r\
    nReceived-SPF: pass (mail.gsa-cs.com: SPF record at _netblocks.
    google.com designates 209.85.128.52 as permitted sender)\r\
    nReceived: by mail-wm1-f52.google.com with SMTP id 5
    b1f17b1804b1-4281c164408so32139285e9.1\r\n for <tika.sah@gsa-cs.
    com>; Wed, 31 Jul 2024 10:15:24 -0700 (PDT)\r\nDKIM-Signature:
    v=1; a=rsa-sha256; c=relaxed/relaxed;\r\n d=gmail.com; s
    =20230601; t=1722446120; x=1723050920; darn=gsa-cs.com;\r\n h=
    to:subject:message-id:date:from:mime-version:from:to:cc:subject
    \r\n :date:message-id:reply-to;X-Received: by 2002:a5d:5144:0:
    b0:368:4c56:2bd0 with SMTP id\r\n ffacd0b85a97d-36
    b5cf243dcmr10318267f8f.33.1722446120274; Wed, 31 Jul 2024\r\n
    10:15:20 -0700 (PDT)\r\nMIME-Version: 1.0\r\nFrom: Tika sah
 <tikasah84@gmail.com>\r\nDate: Wed, 31 Jul 2024 23:00:08 +0545\r\
    nMessage-ID: <CAMUvmDSOUHqDYSAE4OojOSz3LQicQdZQjEu+
    TvrRcFPNYPszGA@mail.gmail.com>\r\nSubject: Test mail for
```

testing model\r\nTo: tika.sah@gsa-cs.com\r\nContent-Type:
multipart/mixed; boundary="00000000000024f8c6061e8e3d8e"\r\n\r\
n--00000000000024f8c6061e8e3d8e\r\nContent-Type: multipart/
alternative; boundary="00000000000024f8c3061e8e3d8c"\r\n\r\n
--00000000000024f8c3061e8e3d8c\r\nContent-Type: text/plain;
charset="UTF-8"\r\n\r\nHi Judy ,\r\nWe are pleased to inform
you that your order 49883468 has been received and\r\nis
currently being processed The items you ordered include 8 EA of
TP25\r\nTorx Plus Driver ID 77558591 1483 We will send you a
separate email with\r\ntracking information once your order
ships In the meantime you can review\r\nyour order details at
your account link if applicable Thank you for your\r\norder\r\n
Best\r\n PioneerLogistics\r\n\r\n-
-00000000000024f8c3061e8e3d8c\r\nContent-Type: text/html; charset
="UTF-8"\r\nContent-Transfer-Encoding: quoted-printable\r\n\r\n
<div dir=3D"ltr">Hi Judy ,
We are pleased to inform you that
your order =\r\n49883468 has been received and is currently
being processed The items you o=\r\nrdered include 8 EA of TP25
Torx Plus Driver ID 77558591 =C2=A01483 We will=\r\n send you
a separate email with tracking information once your order
ships =\r\nIn the meantime you can review your order details at
your account link if a=\r\npplicable Thank you for your order<
br>=C2=A0Best
=C2=A0PioneerLogistics<=\r\n/div>\r\n\r\n
--00000000000024f8c3061e8e3d8c--\r\n--00000000000024
f8c6061e8e3d8e\r\nContent-Type: application/vnd.openxmlformats-
officedocument.wordprocessingml.document; \r\n\tname="test_mail.
docx"\r\nContent-Disposition: attachment; filename="test_mail.
docx"\r\nContent-Transfer-Encoding: base64\r\nContent-ID: <
f_lza3uot00>\r\nX-Attachment-Id: f_lza3uot00\r\n\r\
nUEsDBBQABgAIAAAAIQDfpNJsWgEAACAFAAATAAgCW0NvbnRlbnRfVHlwZXNdLnhtbCCiBAIooA
\r), b')']

Result after extracting content: As the extraction of email content is perform using RFC format , we can easily extract email subject body and sender details.

```
Sender name : Tika sah
Sender email : tikasah84@gmail.com
Subject: Test mail for testing model
Attachments: ['test_mail_.docx']
Body:
Hi Judy ,
We are pleased to inform you that your order 49883468 has been
   received and is currently being processed The items you ordered
   include 8 EA of TP25 Torx Plus Driver ID 77558591 1483 We will
   send you a separate email with tracking information once your
   order ships In the meantime you can review your order details at
    your account link if applicable Thank you for your order
 Best
 PioneerLogistics
```

4.2 Outputs from model on Various Scenarios of email content

The model performance was tested by feeding it with different kinds of data to analyze it. It seemed to work fine on some datasets; it predicted a large number of entities from the emails. However, after its exposure to a more sophisticated set of emails, not all the entities could be extracted. A brief explanation for all these scenarios of the model is shown here:

1. **Scenario 1:Best Case Scenario:**

   Example 1

   In the best-case scenario example 1, the model successfully identifies the relevant entities within the email text. For instance, in an email containing a purchase order, the model correctly extracts the Sender , Receiver , item ID, quantity, unit of measure, cost, order number, and product name without any repeatation of data. The email details are accurately cleaned, saved in the database, and displayed in the UI.

Email Text

"Hello random We must regrettably cancel the order numbered SRM3720109 involving 32 units of Self Lubricating Wear Plates Item ID 03221413 priced at 609 per Each We apologize for any inconvenience this may cause and ask for your understanding Please confirm the cancellation at your earliest convenience and inform us if further steps are necessary Sincerely NextGen"

Predicted Entities

- **Sender**: nextgen
- **Receiver**: random
- **Item ID**: 03221413
- **Quantity**: 32
- **Cost**: 609
- **Product Name**: plates
- **Unit of Measure**: each
- **Order Number**:srm3720109

Example 2

In the best-case scenario example 2, the model successfully identifies some of the relevant entities within the email text. For instance, in an email containing a purchase order, the model correctly extracts the item ID, quantity, unit of measure, cost,and order number without any repeatation of data.However sender was not able to detect by the model. The email details are accurately cleaned, saved in the database, and displayed in the UI. Additionally, the text from DOCX attachments is extracted and stored correctly, contributing to the comprehensive data available for analysis and decision-making.

Email Text

"hi ups , we are cancelling our order 4886260210 for 8 ea of slotting item id : 86820016 . please issue a refund of $ 460 . 67 . best regards , amazon"

Predicted Entities

- **Sender**: Not Found
- **Receiver**: amazon
- **Item ID**: 86820016
- **Quantity**: 8
- **Cost**: 460.67
- **Product Name**: slotting
- **Unit of Measure**: ea
- **Order Number**:4886260210

2. **Scenario 2: Worst Case Scenario** In the worst-case scenario, the model fails to accurately identify and classify the entities. This can result in incomplete or incorrect data being saved in the database, which can mislead decision-making processes.

   Example 1

   While providing complex email to the model , model was not able to provide all entities correctly.In this example we can see that 3 major entities were not classified due to complexity in email content.

   Email Text

   Dear Emily Brown,I hope you are doing well. I am writing to place an order for 120 units of Ergonomic Office Chair with Item ID having EOC45678 under Order Number ORD54321. The total cost for this order is $12,000, and the items should be delivered to 456 Corporate Blvd, Suite 300, Los Angeles, CA 90017. Please ship the chairs in Pallets Unit of Measure: Pallet and ensure that the shipment is handled with care to avoid any damage. The order is being placed by Michael Lee, who can be contacted at michael.lee@company.com for any clarifications.Best regards,Michael Lee"

   Predicted Entities

   - **Sender**: Not Found

- **Item ID**: 03535132
- **Quantity**: 120
- **Unit of Measure**: Not Found
- **Product Name** : Not found
- **Order Number**: ord54321
- **Cost**: 1200

Example 2

While providing complex email to the model , model was not able to provide all entities correctly. In this example receiver value was not classified and also unit of measure value was classified as product name . This is not good result predicted my the model .

Email Text

*"We are pleased to inform you that we have received a purchase order from **RCG** . Attached to this email, you will find the purchase order document in PDF format for your reference. We are requesting **15 pk** of **paper cup 03535132** with price **14.2** with order purchase order number: **4517572516**."*

Predicted Entities

- **Sender**: RCG
- **Item ID**: 03535132
- **Quantity**: 15
- **Unit of Measure**: Not Found
- **Product Name** : pk (Wrong)
- **Order Number**: 4517572516
- **Cost**: 142

4.3 Performance Metrics Comparison

This classification report provides the performance of the fine-tuned BERT model in extracting key entities from email content. It provides the precision, recall, f1-score, and

support of each entity type and the overall averages. Here's a rundown of what every metric means and how they are interpreted:

Table 4.1: Classification Report

	precision	**recall**	**f1-score**	**support**
Cost	1.00	1.00	1.00	12221
Item_ID	0.99	1.00	1.00	18589
Product_Name	0.90	0.94	0.92	8516
Unit_Of_Measure	1.00	1.00	1.00	4015
order_num	1.00	1.00	1.00	22229
quantity	1.00	0.98	0.99	4187
receiver	1.00	1.00	1.00	4000
sender	1.00	1.00	1.00	9180
micro avg	0.99	0.99	0.99	82937
macro avg	0.99	0.99	0.99	82937
weighted avg	0.99	0.99	0.99	82937

1. **Precision:**

   Precision is a key performance metric in classification tasks, particularly in scenarios like entity extraction, where the goal is to identify specific items within a dataset.Precision is the ratio of correctly predicted positive observations to the total predicted positives. It tells us how many of the items identified by the model as a particular entity are actually correct.

   Interpretation:

   - For most entities like Cost, Unit_of_Measure, quantity, sender, and receiver, precision is 1.00, indicating that the model has identified these entities with perfect accuracy.
   - The precision for Product_Name is slightly lower at 0.90.This value would indicate the model actually performs pretty well in identifying the product names with an accuracy rating of 0.90. This means that 90% of the instances identified as Product_Name were correct. This is a good score, reflecting the

fact that the model generally performed well in detecting product names.The remaining 10% are false-positives, cases in which the model has mistakenly labeled other tokens as product names. This can give rise to a myriad of mistakes in data extraction and downstream analyses.

2. **Recall:** It is also known as sensitivity or true positive rate, is a critical metric in evaluating the performance of classification models, particularly in tasks such as entity extraction.It is the ratio of correctly predicted positive observations to all observations in the actual class. It measures the model's ability to find all relevant instances of an entity.

   Interpretation:

   - **Perfect Recall(1.00):** The recall is set at 1.00 for entities such as Cost, Item_ID, Unit_Of_Measure, sender, and receiver. This implies the model has been able to identify all relevant instances pertaining to the entities. A perfect recall score implies the model is very good in recognizing these entities, making it very reliable when used in applications demanding complete identification.
   - **High Recall:** For Product_Name, recall is 0.94, showing that model managed to identify 94% of all product name instances. Quantity entity yield is reported at 0.98—meaning that the model really missed a couple of instances. Though active, the assessment shows that the model missed some instances. This indicates that the way the products are named might not be consistent or that it may be over-important specific in some cases. The high recall in this score means that the model knows how to pick out quantities, meaning the model is very well-designed for this particular type of entity.

3. **F1 Score:** The F1-score is the weighted average of precision and recall, providing a single metric that balances the trade-off between the two. It is especially useful when you need to take both false positives and false negatives into account. In general, F1-score is a harmonic mean that combines precision and recall into a single metric. It provides a balanced evaluation of the model's performance, considering both false positives and false negatives.

Interpretation:

- **Exceptional F1-Score (1.00):** Most, such as Cost, Item_ID, Unit_Of_Measure, order_num, sender, receiver, and quantity, have the F1-score of 1.00.This indicates that the model is performing exceptionally well in identifying these entities, with both high precision and recall. This means that the model systematically identifies the entities without any loss, either in terms of precision or recall.
- **Excellent F1-Score:** The quantity has an F1-score of 0.99, which still reflects excellent performance, with only a minor trade-off between precision and recall.Such a high score suggests that the model is almost going to be perfect at work on the quantities front, with just a minuscule compromise between the precision and the recall. This is all about saying that an effective model will catch the specific quantities, tracking the wrong bins very low on both false positives and false negatives.
- **Good F1-Score:** The F1-score for the Product_Name entity is 0.90. This score is the product of slightly lower precision measures (0.92) and recall for the entity (0.92). While the model does very well in correctly classifying product names, there is scope for improvement on consistently identifying all instances and not misclassifying other tokens as well.

4. **Support:** Support refers to the number of actual occurrences of each entity in the dataset used to evaluate the model's performance. It provides information about the frequency of each entity within the dataset.In general , Support refers to the number of actual occurrences of each entity in the dataset.

   Interpretation:

   - The order_num entity has the highest support, with 22,229 actual occurrences in the dataset. It means that order_num is the most frequent entity. This is important for both high precision and recall of the model with this entity because it will probably most strongly affect the performance and reliability of the model if there are mistakes in the identification of such a common entity.

- The product name entity is very well represented, with 8,516 real occurrences in the dataset.It means that this relatively high frequency of product names in the dataset partially explains the effect of slightly lower precision and recall for this entity on the overall metrics. Given the frequency of this entity, this can result in errors of product names tagging to a greater extent, leaving space for further improvement.
- Entities like Unit_Of_Measure and quantity have only 4,015 and 4,187 actual occurrences, respectively. While these entities are not as frequent in this dataset compared to order_num and Product_Name, the model appears to handle them satisfactorily based on the precision and recall values. Since these entities appear at a lower rate, their misidentification would impact the performance measures less significantly.

5. **Micro Average:** The micro average is a method for calculating performance metrics by aggregating the contributions of all entities to compute a single average metric. This approach is particularly useful for understanding the overall performance of a model, especially in scenarios involving imbalanced datasets.

   Interpretation:

   - The micro average precision, recall, and F1-score are all 0.99, indicating that the model is highly effective overall across all entities. This suggests that your model performs consistently well, even when all entity predictions are considered together.

6. **Macro Average:** The macro average is a method for calculating performance metrics by computing the metric independently for each class (or entity) and then taking the average. This approach treats all classes equally, regardless of their frequency in the dataset.

   Interpretation:

   - The macro average precision, recall, and F1-score are slightly lower at 0.99, 0.99, and 0.99, respectively. This might be a small difference from the micro average, which could have been caused by poor performance on the

Product_Name entity type, therefore making the model less consistent when evaluated equally across all entities.

7. **Weighted Average:** The weighted average is a method for calculating performance metrics that takes into account both the support (frequency) of each entity and the individual metric scores (precision, recall, and F1-score) for those entities. This approach provides a balanced view of model performance, reflecting the importance of each entity based on how frequently it appears in the dataset.

   Interpretation:

   - With weighted average precision at 0.99, recall at 0.99, and F1-score at 0.99, the model demonstrates a well-balanced and effective performance across the most common entities. This close alignment with the micro average indicates that the model is robust and reliable, making it a strong candidate for information extraction tasks. By monitoring the weighted average, one can ensure that the model continues to perform effectively, accounting for the importance of frequently occurring entities in the dataset.

Overall, your BERT model works out pretty well for extracting key entities from email content, with much percentage of precision, recall, and F1-scores close to unity for most of the entities. A little bit of underperformance for the Product_Name entity makes some sense; this could be one of the trickier entities, as there might be some variety in the ways product names are styled in emails. The last two confirm that the model shall, in general, be robust and reliable, with very high accuracy on average. This classification report is a pretty strong validation for your model while executing the task of information extraction from the emails.

The confusion matrix is convenient to verify the performance of our information extraction model, providing a detailed view of how well the model is classifying each type of entity. The confusion matrix shows the number of true positives, true negatives, false positives, and false negatives for each class, which is allowing one to calculate the accuracy, precision, and recall and correspondingly the F1 score of the model. Through

the analysis of these metrics, we will learn about the strengths and weaknesses of the model and areas for further improvement.

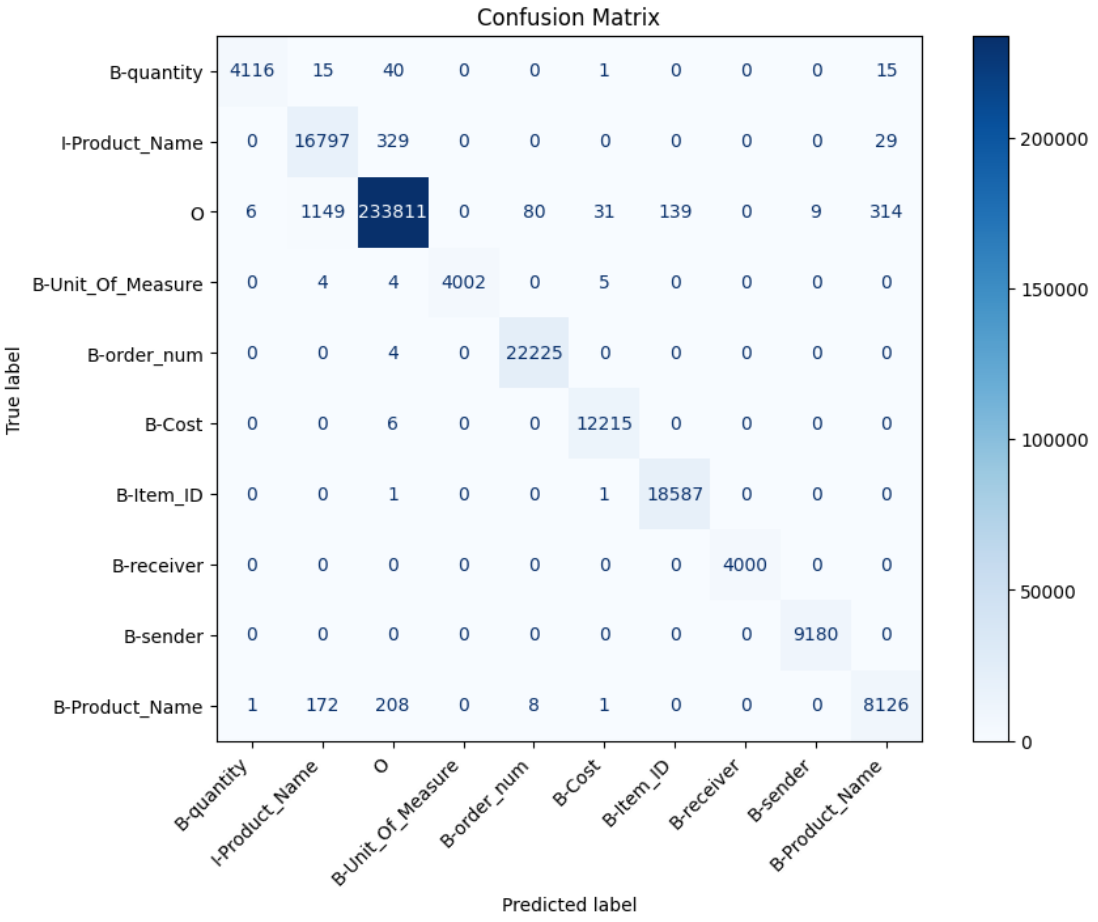

Figure 4.1: Confusion Matrix

4.4 Loss Curve Analysis

The Figure shows the loss curves for training and validation loss across five epochs which is trained over 20000 data.A loss curve is a graph that shows changes in the model's loss over time through training. This graph shows the relationship between the loss and the number of epochs, which are iterations through which the whole dataset passes. These curves help diagnose the training process and show how well the model is learning from the data. These general trends in the curves already reflect that the model is indeed learning and generalizing to unseen data.

- **Training Loss:** The blue line represents the loss calculated on the training dataset. This is the data that the model learns from during the training process.
- **Validation Loss:** The orange line represents the loss calculated on the validation

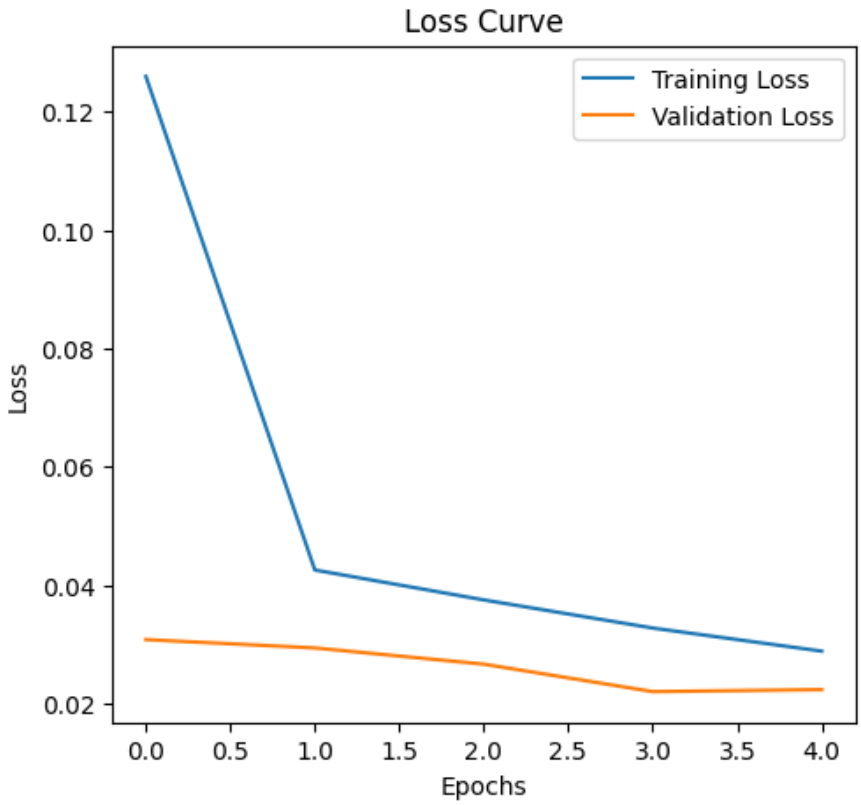

Figure 4.2: Loss Curve

dataset. This dataset is not used for training but for evaluating how well the model generalizes to unseen data.

- **Loss:** Loss is a measure of how well (or poorly) the model's predictions match the actual labels. Lower loss values generally indicate better model performance.

Analysis: The curve of loss versus time shows how the model learns over time; it starts high at epoch 0, which is quite normal because the model has just started to understand the training data. During training, there is a large drop in loss, especially at the beginning from epoch 0 to 1, where the training loss drops drastically from about 0.12 to 0.03, indicating that essential patterns are being learned rapidly. On the other hand, the validation loss is lower than the training loss at the beginning and decreasing; this means that the model learns to generalize beyond the training dataset. Between epochs 1 through 4, the training loss indeed does continue to fall, although at a slowing rate with a curve starting to flatten at around 0.02 by epoch 4, proving the model fits the training data quite effectively. The trends of validation loss are all similar: first, it slightly decreases and then levels out. This already put a clue that the model learns well,

and not only has good generalization toward the validation set. Already this pattern of loss behavior suggests that the model is getting closer to optimal performance but is seeking a good balance between fitting the training data and generalizing to new unseen data.

Observation: The loss curves indicate overfitting not to be a major issue here. Usually, overfitting is indicated by a rapid drop in training loss with no obvious decrease, or even an increase, in validation loss. In this case, though, both the training and validation losses go down and then stabilize together. There can hence be said to be very well-balanced model performance across both datasets. Moreover, the flattening of both loss curves towards the end of training may indicate convergence—that is, a point at which further training is unlikely to yield substantial improvements in performance. The curve also indicates that the model was trained for 5 epochs, which seems appropriate while both losses are stabilization and going further may result in overfitting without actual improvement.

4.5 Accuracy Curve Analysis

An accuracy curve is a critical tool for understanding how well a model is performing during training and validation. It plots accuracy against the number of epochs, showing how the model's predictive power improves over time.The accuracy curves shown on Figure represent the Training Accuracy and the Validation Accuracy over the five epochs. These curves give an indication of how well the model was performing by getting the labels correctly.

- **Training Accuracy:** The Blue line curve shows how well the model is performing on the training dataset. It indicates the proportion of correctly predicted labels out of all predictions made on the training data.
- **Validation Accuracy:** The orange line curve shows the model's accuracy on the validation dataset, which was not used in training but is used to assess how well the model generalizes to unseen data.

Analysis: From epoch 0 to 1 in the first stage of training, the model improves its training

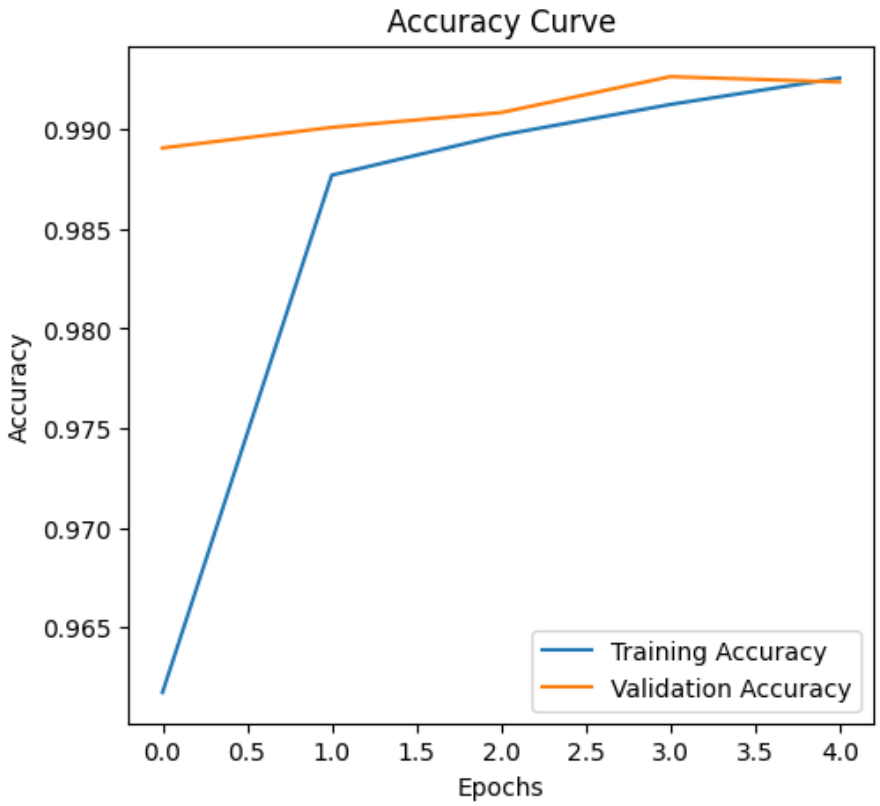

Figure 4.3: Accuracy Curve

accuracy from about 96.5% to 98.7%. This very initial rapid rise during training is a very good indication that the model is learning the underlying patterns in the training data. The validation accuracy actually starts higher than training accuracy, beginning at about 98.9%, which may indicate some sort of bias or variance in the beginning of training. This discrepancy is, however, rectified quickly as the model corrects its learning in the following epochs and comes closer to the validation performance.

In the middle phase, from epoch 1 to 3, the training accuracy continues to increase but at a decreasing rate, finally reaching around 99.1 percent at the end of the third epoch. This could be a gentle improvement in the model's prediction based on training data. Meanwhile, the validation accuracy is slightly growing and converges to a value of about 99.2%. These early indicators are very positive in regard to the improved generalization capabilities with respect to the model. At the last phase, from epoch 3 to 4, both training and validation curves have flattened, reaching about 99.3% each. This plateau probably indicates that the model has learned enough from the training data; additional optimization is not possible. This step, however, proves that the training and

validation accuracies are very close, thus the model generalizes well without overfitting since it performs almost the same on both sets of data.

Observation: The fact that the validation curve is very close to the train curve means that there is very high generalization toward unseen data for the BERT model, with no huge gap suggesting overfitting. This good generalization ability is further enhanced by both stability curves at the end of training, thus proving that the model has reached its optimal accuracy and additional epochs may not bring further improvements. It shows that this large accuracy increase at the first epoch means the model has learned to quickly identify important patterns in the data, thus effectively learning early and therefore contributing to overall robust performance throughout training.

4.6 Overall system in Web App:

System for extraction of information from email is build in Django Framework with clean user interface which can be understand by any common man easily .All the process and important information is displayed on user interface for the enduser to evaluate . Each step that is performed by the system is explained and image for result in web app are below:

The workflow of fetching new, unseen emails is driven by email credential input. With these credentials, it connects to the email server using IMAP technology for authentication. Once authenticated, Celery, a distributed task queue, takes care of the task to check the email server for new messages asynchronously. This is configured to run at regular intervals to keep the application current with the latest email data by polling continuously for new emails. On detection of new, unseen emails, Celery downloads these messages and queues them for further processing. In this way, the main application does not get interrupted due to this asynchronous approach; Celery efficiently fetches emails. The integration of Celery with Django makes it easy and clean to handle and monitor these background tasks using the Django admin interface. Administrators have tools for tracking email fetching status and logs.

After fetching the emails, relevant content has to be extracted from the email body and

attachments. Parsing of the email body takes place for textual content, which is then prepared for processing. Mostly .docx attachments are downloaded and processed for extracting their textual content. Extracted text from these will be cleaned of formatting issues, special characters, and irrelevant data. This cleaning step itself is quite important for standardizing the text, removing HTML tags, normalizing white spaces, and finally converting the text into a uniform case. The cleaned content would then be stored in the database for further processing.All the new email will be displayed on the dashboard with details of email .After Fetching details of email is shown on the dahboard of the Web App that is shown in image.

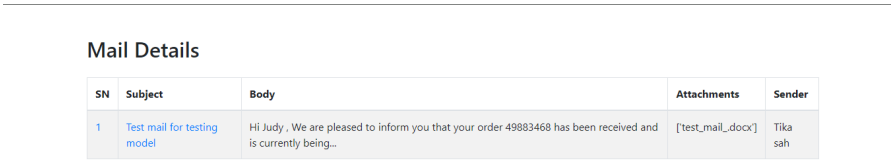

Figure 4.4: Dashboard of Web App

The cleaned text content from the e-mails and attachments is then saved into a structured format within the Django application's database. This structured storage will allow later access and further processing of that data efficiently. The database design is done in a way that it is possible to store all information, including the mail content, meta information of the mail like sender and subject, and also the text which is extracted from attachments. Each record is associated with the relevant email for proper management of data. This is also managed by Celery for asynchronous and efficient updating of the database. Data that are store in database in structured format are shown in figure below:

The content now stored in the database is ready to be fed into the BERT model for processing. The BERT model, which is state-of-the-art in natural language processing, is specifically fine-tuned to recognize key entities within text. When the content text is passed into the BERT model, it extracts entities such as order numbers, item IDs, quantities, sender, receiver, and any other relevant information. These extracted entities will be saved back into the database, linked to their original email record. Finally, the

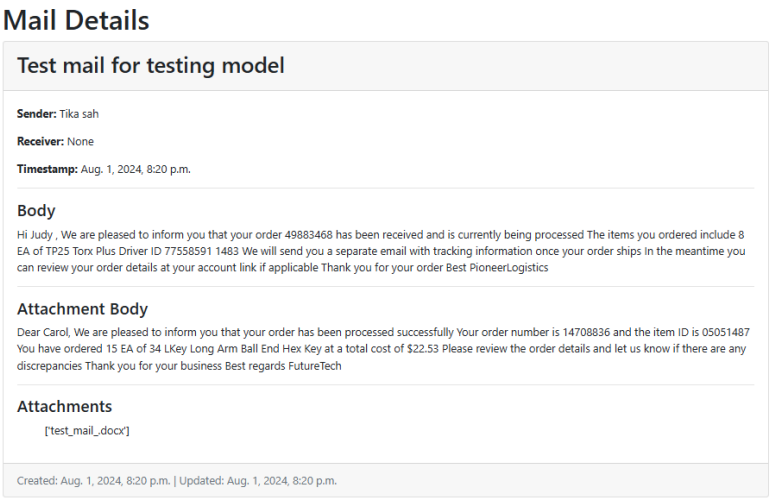

Figure 4.5: Details of Email

dashboard will present all the extracted entities in a user-friendly manner. This provides users with an organized view of all key information extracted from emails and their attachments. Users can filter, sort, and search through the data to find specific details. The results are then integrated into the Django web application, where the dashboard offers an easy user experience for accessing and managing all relevant information efficiently. The result of the model stored in database are show in web app graphical user interface like :

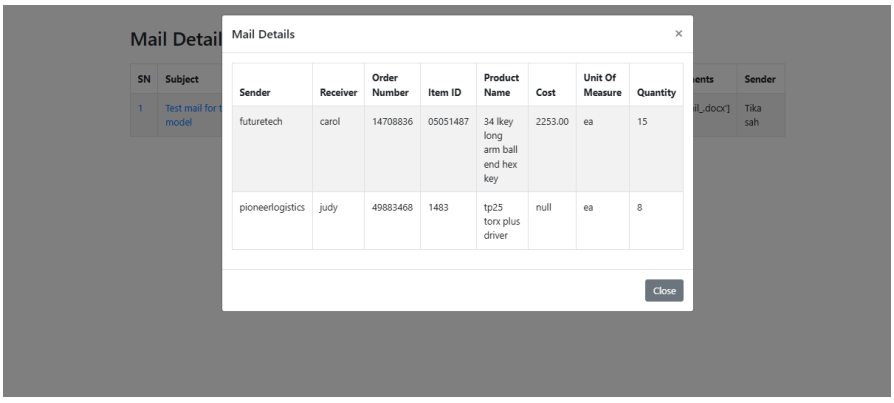

Figure 4.6: Result from Model for User interface

The Django web application provides a smooth, user-friendly way of doing this through the automation of the extraction and the very well analysis of email content and its attachments. The processing of extracting key entities both from the email bodies and their attachments is further eased at more advanced levels by such tools as Celery and BERT.

This system means end users have a unified platform from where they can access all the related information easily, in one place. The interface is easy to use and can help a user manage and monitor the content of emails, view the entities extracted, and conduct searches or apply filters to the extracted data. This centralized technique augments operational proficiency and enhances business capability by way of offering a generalized view of indispensable information.

This web application is important to business processes as it automates workflows that currently have manual emailing activities, thereby making data more accessible. Therefore, it is a valuable tool useful to any business interested in increasing efficiency for email management and information extraction.

5 DISCUSSION AND ANALYSIS

5.1 Comparison of Theoretical and Simulated Outputs :

5.1.1 Theoretical Expectations:

Theoretically, email content extraction was to be quite efficient and straightforward, wherein extraction from any email could be effected seamlessly, regardless of the security protocols in place. Extraction procedures were foreseeably uniform across all email platforms, ensuring that no security concerns impede the retrieval of content. This would make the system accept emails from different sources without running into major obstacles related to encryption, authentication, and access permissions.Furthermore, it was expected that there would be easiness and speed when it came to extracting content from attachments to emails. Extracting text from attachments, be it in PDF, Word, or other file formats, was supposedly done with little effort and not by special methods. It was to be efficient enough to process a large number of attachments without much loss of time. This was all based on the assumption that extraction tools and techniques would be developed versatile and robust enough to embrace file types by treating them alike, and thus not requiring special extraction strategies or additional processing time.

Also the model will be very accurate in extracting relevant entities from email content. Literature explains that BERT, or Bidirectional Encoder Representations from Transformers, goes well on a variety of NLP tasks, especially contextualizing and getting the subtle differences in text.Due to which with fine tuning the pre trained Bert model will provide us high efficiency for the model to extract key entities form the text content gathered from email body as well as from attachments.It was expected that BERT would particularly excel at tasks such as sender, receiver, order number, item ID, and other relevant information extraction from a broad and complex set of structures in emails. The expectation is also forwarded by the fact that BERT is pre-trained on large corpora and can fine-tune with a specific task in a manner such that it would adapt to the oddities of the email domain. From theoretical models, it is expected that with sufficient fine-tuning on some domain-specific dataset, BERT shall provide high accuracy in entity extraction, very much at par with state-of-the-art in similar NLP tasks.

The theoretical expectations were also informed by a comparison with state-of-the-art

methodologies in the field of information extraction and named entity recognition (NER). Studies have shown that transformer-based models like BERT outperform previous approaches, such as recurrent neural networks (RNNs) and conditional random fields (CRFs), particularly in capturing long-range dependencies and understanding the context within a document. These findings supported the expectation that the system would deliver superior performance compared to earlier, less sophisticated methods.

5.1.2 Simulated Outputs:

The system was set up to run with a less secure email account using IMAP, which is a kind of Internet Message Access Protocol technology. IMAP shall be used because it allows an email client to access and manipulate messages stored on a mail server, hence suitable for real-time email processing. However, because of the reduced security, less secure nature of the email account meant some security concerns mitigated and thus the system could bypass some of the common authentication and encryption hurdles normally encountered with more secure email accounts. This choice facilitated testing but it also drew out the potential vulnerabilities in the system when handling more secure environments. The use of less secure accounts was basically for ease in the implementation process and majorly focusing on the core functionality, which is content extraction; this may not fully be indicative of complexities encountered in more secure settings.

The main two primary components extracted in the extraction process were the body of the email and its attachments. Text is extracted from the email body, so it easily provides a mechanism for retrieving the central piece of information that is richly conveyed in the email. On the system, plain body emails can effectively be treated, and organized content within the body of the mail is parsed and prepared for processing by the BERT model.Simulation was done with attachments of type .docx only. The reason for selecting .docx files is that these are fairly common among attachments, and the content within them is a little bit structured. The system was able to extract text from such .docx attachments in an efficient manner so as to ensure that the content was in a form that was appropriate for analysis. The extraction of .docx files went relatively smoothly; the system would parse and extract relevant text files housed from within the relevant documents. Important in being testable is whether the system could work on data sources

within emails by extracting data without limiting the source to only the body of the email.

The BERT model has been used here because it is capable of providing contextual information throughout the process of information extraction. Fine-tuning the BERT model for this application generated quite impressive results in regard to the identification and extraction of these key entities from email content. Deep understanding of the context is where its strength lies, which helps this model in identifying and extracting relevant information from complex or nuanced text accordingly.

The simulations, however, showed that the performance of the BERT model was significantly affected by both data quality and diversity during the training phase. Therefore, generalization or good performance on unseen data by the model is pegged on the examples exposed to it during training. If there is not enough variety within the training dataset for instance, in the structure of emails, kinds of languages, and sorts of attachments the model may end up poorly performing at the processing of emails that break the patterns it was trained on. Therefore, the success of the BERT model in such an application will be as good as the comprehensiveness of the dataset used in training it. It will require a diverse and representative view of data so that the extraction of information could be done in a robust and reliable way across scenarios.

5.1.3 Analysis of Discrepancies :

The mail content extractor system had a number of discrepancies that were affecting its performance and security, mostly with respect to the highly secure electronic mail services like Gmail. Some services, such as Gmail, have very strict security whereby it's not possible to access them directly using passwords from code. This poses quite a challenge for any automated systems that strive to fetch email content. That meant the system had to run in an unsafe mode—a mode with a number of security features disabled to allow access. While doing this eased the extraction process, it brought along big risks to security. Running systems in less secure modes opens them to possible security breaches, something not quite to the style wanted for environments where data privacy and security are paramount. This already points out the dilemma between the ease of implementation and the robustness of security standards within an email processing

system.

Another big challenge in the flow was extracting text content from email attachments, mostly PDFs. Extracting text from a PDF turned out to be quite time-consuming, particularly in cases where they had image-based content. Most tools that extract text usually have problems with picture-rich PDF documents, for they require OCR processes to turn images into texts. This added a level of difficulty and further decreased efficiency in the extraction process. Such problems with the system restricted it to work only with .docx files, since they are usually easier to parse out for text content. Performance issues could be seen even for .docx files when the documents contained embedded images. The presence of images inside a .docx file usually slows down the extraction process, since it handles the text and image data together, thus putting more load on the system resources and reducing overall performance. The most striking deviation was the one between the theoretical expectation of the model and the receipt of it with regards to the BERT model. Theoretically, the BERT model was expected to provide strong results since it was to understand the context and extract information of relevance. The model's performance was below par. This difference arose from the fact that diverse training data was not fed into the model. For it to generalize well across various types of email content, the BERT model needs to be trained on diverse data that includes a wide range of email structures, linguistic styles, and different content. Unfortunately, diversity in emails was limited within the scope of the dataset available for training, which had only a narrow range of email types. This model did not generalize due to the lack of variety, fitting only certain types of emails on which it had been trained. When given new or different types of email content, the model's performance decreased as it was not exposed to enough examples during training.

This analysis revealed that, although the system might have been operable under some conditions, it was much less resolute in meeting the broader and more diversified requirements needed for robust performance. The security concerns, challenges of attachment processing, and the limitations of the training data all contributed to discrepancies between the theoretical model and real-world performance. These findings underline the fact that, in case of system versions aimed at higher security and performance along with

generalized email content extraction functionality in the future, such challenges will have to be addressed.

Theoretically, a BERT model should be able to identify and classify the sender correctly, item ID, quantity, unit of measure, cost, and order number from e-mail content and .docx attachments. The theoretical expectations were that with enough training data and proper tuning, it would approach perfect precision and recall across all entity categories. Although most of the simulated outputs had been quite accurate, there were still noticeable discrepancies with the identification of entities such as ”Product Name” and ”Cost.” The model at times couldn't correctly classify or misses entities due to their expressions that vary within the emails. These disparities can be attributed to differences in the use of language within the emails, the intrinsic difficulty of the texts, and the representation of entities within the training data. Although it performed well for most, BERT performed really excellently on some entities but poorly on others where the training data did not represent variations that exist in real-world email content.

5.1.4 Error Analisys:

1. **Email Access:** To access email content directly from the system with easy access . The accounts had been set up with less secure settings using IMAP—Internet Message Access Protocol—the technology applied to access email messages in a mail server locally from a client—and were used during the test and implementation phase of the email system. However, these accounts were set up without considering modern security standards. As a result, they are reliant on basic or legacy authentication. To better protect data security and privacy, most modern email service providers, such as Gmail, Outlook, and Yahoo, implement stringent safety measures that result in the limitation or total blocking of access to these simple username and password combinations. Most of these providers moved to more solid security frameworks, a prime example being OAuth 2.0. OAuth 2.0 is an authorization framework that gives third-party services the ability to exchange access tokens without user credential exposure. It improves security by providing limited access that is also revocable to resources, not like the traditional ways where the credentialing is static and thus open to exploits.

   To use less secure email settings during testing and deployment, has broader

implications on security and functionality. Modern authentication techniques like OAuth 2.0, with configuration of email accounts at par with current security standards, become very important in mitigating this risk. This will improve not only data protection and regulatory compliance but also provide a consistent level of access to email content, enable the use of advanced features, and ensure that the email system is more reliable and secure.Due to lack of accessing only less secure email configuration it's is complicated to connect any types of mail instantly with being configured to less secure mode.

2. **Text Extraction From Attachments:** One of the major problems during the development process of the email system was the way to handle different types of content in attachments of emails. Attachments could include PDF, DOCX, image, spreadsheet, or any other format; all these have their own structure and type of content. The system, however, was truly aimed at the extraction of meaningful text content to automate information processing. As an initial step, the system tried to process all kinds of attachments indiscriminately, with mixed results. It turned out to be pretty hard to extract text from complex or non-text attachments like images, charts, or scanned documents, usually giving incomplete or inaccurate data extraction. It also introduced unnecessary complexities in a system, along with processing overhead, since it had to deal with different file formats and extract content which is not relevant for the task in hand.

   Challenges continued to remain embedded inside the very DOCX files themselves, sometimes containing non-text elements like images, tables, and embedded objects. The system should have been designed to extract the text content of such documents while ignoring images and other non-textual elements. This selective approach ensured that the data extracted would be relevant and useful for the intended purpose; it also made the process of extraction easier. However, despite these improvements, the system faced some limitations during the handling of DOCX files with mixed content. For example, images and all other non-text elements in the DOCX document were not extracted. It currently extracts text information from attachments only and therefore captures no extra detail that may manifest in the form of pictures or diagrams. Having cut down focus to DOCX attachments and chosen text content, it resulted in a much more efficient and directed way of

text extraction. This refinement reduced unnecessary complexity and increased the accuracy of the extracted data. Processing images and other non-text elements within a DOCX file remains outside the realm of possibility, and this limitation does point toward an area of future improvement.

3. **BERT Model Traning and Performance:** The BERT model itself is very powerful in natural language processing tasks but is significantly based on the quality and diversity of the training dataset. Its ability to understand and generate text is thereby based on the exposure to examples covering diverse possibilities while training. In this case, performance was limited due to the narrow and limited variety in the training dataset, mostly filled with a certain type of email content.

   Basically, overfitting means the model becomes excessively biased to the specific patterns and features of the training data alone, such that it fails significantly on new, unseen data. The BERT model fit the training dataset well but didn't generalize to any other kind of email content since the diversity in the dataset was less. Overfitting in particular is especially a problem in real-world applications where the input data is likely to be very different from the training data. The ability of the model to apply what it has learnt from the training dataset to new unseen examples is referred to as generalization. This meant due to the limited diversity of the training dataset, this resulted in the BERT model having a very narrow understanding of email content, poor generalization. In cases where the structure, subject, or linguistic nuance differed from what it was trained for, the model was not good at extracting the required information. This weakness pulls down the effectiveness of the model in handling real-world scenarios, wherein e-mail content may be hugely different across format, language, and context.

   There are two most important approaches to mitigate the narrow training dataset: augmenting a training dataset and applying data augmentation techniques. Expanding this dataset to hold a greater variety of emails is significant for good generalization performance. It would involve including emails from different sources, industries, and purposes. For example, the dataset may include emails

from sectors with specific jargon and styles of writing, such as finance, health, retail, and technology. If it is expected to work with multilingual content, the training dataset should contain emails written in multiple languages and dialects. This will enable the model to learn the linguistic variations and process them, hence making it more versatile in any real world application. There should also be preparation for the model with any kind of format for emails—for example, being structured with tables, bulleted lists, images, and attachments—all within the dataset. It trains the model to navigate and extract relevant information from emails, independent of their formatting. Expanding this dataset within a broad range of contexts—from customer support and marketing to internal communication and much more—enables the model to learn handling situations. This will make it more robust when handling real-world use cases with varied email content. Data augmentation means generating additional examples by small perturbations in existing examples. For example, training data with augmented instances in which words are changed in emails, synonyms used, or certain phrases rephrased can be used. This exposes the model to a variety of language that it may encounter in real-world emails. Augmentation may also change the structure of the training emails. For example, emails might be reformatted in different layouts or with added/removed content such as headers or signatures. This will teach the model to focus on the core content, regardless of superficial changes in structure. Addition of noise in the form of typos, abbreviations, or informal language can further improve model robustness. This prepares the model to deal with imperfect data, quite common in real-world email communication. Further augmentation of the dataset should be done by simulating different scenarios like levels of formality or urgency of e-mails to increase understanding of context by the model. This makes it very effective in correctly interpreting the intended meaning and extracting relevant information in diverse situations.

5.2 Comparison with State-of-the-Art Work :

Our information extraction project focused on extracting named entities using Named Entity Recognition from email data. The results were much better compared to what Padmanandam et al. achieved for the CourseNetworking platform in their 2024 paper,

using a custom-made NER model. This could be because though both were based on BERT-based models, the context, objectives, and outcomes differed.Padmanandam et al. developed a BERT-based NER model for extracting skills from user profiles in the CourseNetworking platform. Their best F1-score on test data was 68%, while the ranges for precision, recall, and F1-scores went from 0.6 up to 0.72. Our project, tasked with extracting 8 different entities from email data, reached accuracy of 98% on a dataset of 20,000 samples.

Padmanandam et al.'s model delivered an F1-score of 68%, hence a fair middle ground for both precision and recall, with metrics lying in the range 0.6 to 0.72. From this, it is deduced that though the model was fairly adequate at the identification of the entities relating to the skill domain, it somewhat fumbled in their proper classification. Although the model was trained with a dataset of about 50,000 entities, it was suffering because of its single type nature skills.Our model, on the other hand, was powered by an impressive accuracy of 98%, which speaks of a very high degree of precision in predictions across multifarious types of entities. Even when the dataset was reduced to 20,000 samples, our model could afford the complexity of extracting and classifying eight different entities, such as sender, receiver, order number, item ID, product name, cost, unit of measure, and quantity. This flexibility and accuracy underline very well that our approach is highly robust and can handle a wide array of classes within the dataset. This comparison epitomizes how strong the model is in raising high performance with respect to many entity types, whereas the model from Padmanandam et al., although potent, was quite specialized and had quite a hard time retaining precision and recall.

Several factors contributed to this model's better performance in information extraction over e-mail data. First, the structured nature of these e-mails, with clear headers, bodies, and attachents text, delineated clearly and offered contextual richness that helped this model in the identification and accurate classification of the entities. This well-defined structure offered an explicit cue for entity boundaries and provided enhanced ability to understand the relationships among different entities. This model was, however, designed for the extraction of eight diversified entity types in contrast to a model focused on one single entity. Thereby, in contrast to Padmanandam et al., because of the use

of a well-balanced and representative dataset, bias could be avoided and generalization fostered. Even the training and thorough tuning—including hyperparameter optimization and transfer learning from pre-trained BERT models—further contributed to the model's effectiveness. Iterative improvement cycles with error analysis and corresponding targeted adjustments were important for refining performance. Structured email data, diversity in the entities, and rigorous training joined at a junction to make this model very accurate in extracting information.

Performance Evaluation : The authors' CN Skillset dataset was based on around 50,000 entities extracted from CourseNetworking user portfolios. This dataset is oriented to detect the mention of skills in educational and professional contexts; it has attributes like the type of skill and mention frequency.[5] A skill had to be mentioned at least twice to be harvested in this dataset, ensuring a certain level of relevance and frequency. Processing on the dataset included sentence tokenization and tagging every word as either "Skill" or "O" for Others. This broad and heterogeneous dataset, tuned to capture skills across different user-generated texts, was instrumental in the training of a model that could handle diverse contexts within professional networking, hence contributing towards its ability to generalize well across different scenarios.

In contrast, my dataset was created from personal email transactions with the goal of extracting a diversity of entity types like product names, costs, and quantities. Although this dataset helped the model perform highly when the test set consisted of familiar email patterns, it did not perform well on new and complicated email patterns. Processing my dataset involved tokenizing the text and tagging it with multiple entity labels specific to email content. This domain-specific approach delivered high performance within the very familiar patterns in the dataset but resulted in poor generalization to unseen email types, thus proving the limitation of a more narrowly focused dataset.

Results obtained from the CN Skillset dataset gave an overall less accurate model with enhanced handling of a wide breadth of contexts and user-generated content. This may be indicative of the nature of the dataset in general, reflecting the robustness of the model to very heterogeneous data. This broader scope of the CN Skillset allowed the model to generalize across a wide range of types of skill mentions; this came at the cost of some lost precision.My model was trained from the dataset of personal emails and worked

with quite a high accuracy within the range of concrete familiar patterns, thus proving the model is fine for known email types. This model fared poorly when presented with new or different email patterns. This is an example of the accuracy-generalization trade-off: my model did wonders within one domain but turned weak towards varied email content. On the other hand, unlike mine, this CN Skillset-based model could be regarded as a generalized approach with applicability in a much broader context.

5.3 Challenges Encountered:

In this project, several issues were encountered during the training phase, particularly in fine-tuning the BERT model. The first and probably most important issue is related to the very early saturation of the model because of the limited dataset variety and complexity. The training dataset used for this BERT model was based on real-world business emails. Although they were real, they were rather non-diverse with respect to content and structural features. The majority of the emails followed patterns and represented routine business communications, such as meeting requests, status updates, and memos. This homogeneity meant that similar examples were redundantly shown to the model during training and the model constantly learned to identify these patterns, rarely exposed to a broader range of entity types or contextual variations.

Challenges in Training BERT for Information Extraction from Emails One of the major issues faced during the course of fine-tuning the BERT model for information extraction from emails was the very early saturation of the model. As is expected with a highly complex, large-scale pre-trained model, BERT is designed to handle most of the variances in data and context. However, some limitations observed in the dataset impacted the effectiveness of training in this project. This diversity was missing in this dataset, which contained emails with similar patterns and formats. The dataset was too homogeneous, and this homogeneity had exposed the model to repetitive examples, which restricted its ability to generalize and adapt to new, unseen data. Since BERT works on a diverse and complex dataset, this uniformity in email patterns degraded the model's performance drastically. Moreover, the dataset was relatively small with regard to the enormous parameter space of BERT. Often, this smaller dataset lacks enough examples to learn from, in which case the model may suffer from overfitting and poor generalization. This would then mean that the saturation in the model had set in earlier during training, and performance on the training set reached a plateau, while improvement on unseen data

became minimal. Therefore, hyperparameter tuning was given a crucial place in the training regime as far as the pursuit of addressing these challenges is concerned. In fine-tuning BERT, one would have to make a change in the learning rate, batch size, and the number of training epochs to re-tune the hyperparameters. An appropriate chosen learning rate will balance convergence speed with stability, while a high or low learning rate could result in inadequate learning or overfitting. Similarly, the optimal batch size and the number of training epochs had to be iteratively arrived at. Actually, all the above modifications were done to optimize the performance of the model under constraint due to a limited dataset. Finding a combination that would work well perfectly in effective learning without leading to overfitting eluded me, hence the massive validation.

To counteract the effect of overfitting and avoid saturation in the very early stages, another strategy was applied by implementing the mechanism of early stopping. Early stopping refers to a regularization technique that stops the model before the process starts showing a downturn in performance improvement on a validation set, thus preventing the model from overfitting. Early stopping ensured training termination when further improvement in model performance on another validation set in the course of training was not observed anymore. This, therefore, makes early stopping limit the risks associated with overfitting and controls the time the model is trained for, hence preserving its ability to generalize. While early stopping did a great job balancing the training time with the performance of the model, it also brought out the relevance of the quality and diversity needed in the dataset for optimal results.

5.4 Overall Analysis:

Overall model result can be analyzed based on the result provided by the confusion matrix on validation dataset with is 20% of overall data which was provided. In this confusion matrix, rows represent the true label of the entities, while columns represent the predicted labels. The diagonal values show correct predictions by the model, while off-diagonal values show misclassifications.For instance, it correctly predicted 4116 cases of B-quantity, 16797 cases of I-Product_Name, and 233811 cases of the label O, meaning not an entity text. However, some miss-classifications are there: 15 of the B-quantity were misclassified as I-Product_Name, while 329 of I-Product_Name were miss-classified into the O category. These errors thus reveal that it is common for the

model to get specific entities mixed up, especially if they are close to each other in the text or when the context is not so clear.Large values on the diagonal of this matrix indicate that the model works generally well, classifying most of the entities correctly. In a perfect model, the off-diagonal values would not appear at all. The presence of these values gives room for possible improvements, oriented mainly to a greater differentiation between entity types and avoiding confusion with the 'O' label.

Thsese challanges can be address by adding more training data, including examples of commonly misclassified entities; applying data augmentation techniques; or simply further fine-tuning the model at hand. All these actions would very likely improve the model's accuracy in identifying the entity boundaries and hence reduce misclassifications. Confusion matrix, in general, shows that even if the model is good at most things, such targeted improvements will enable it to perform better in real-world extraction tasks.

Some very promising ways to further improve this NER model are by improving the dataset itself, making the model robust and generalizing well. The dataset may currently not be very diverse, thereby poorly recognizing a wide range of named entities and classifying them. Additional datasets that can be included add more variability from other domains, industries, and geographical regions to expose the model to a wide-ranging spectrum of data. These can include mails from other sectors, modes of communication, document types, such as customer support tickets and corporate reports. This will not only handle the sparsity in the data but also go a long way to mitigate class imbalance by adding examples with rare or infrequent entities. Data augmentation techniques can dramatically improve the model as well. Synthetic variations of existing data increase the size of the training set effectively. A good number of these techniques, such as entity swapping and paraphrasing, require making a diverse set of training examples by replacing target entities with their synonyms or rephrasing sentences without changing their meaning. One can further improve it by introducing controlled noise or slight distortions in the text to make it more resilient against real-world variations. All these techniques overcome limitations of the data and class imbalance problems, finally giving a more resilient and adaptable model.The other important aspect is advanced hyperparameter tuning for model performance optimization. Given the complexity of BERT and specific requirements for NER tasks, careful tuning for hyperparameters like

learning rate, batch size, and dropout rates should be considered. Systematic methods of grid search, random search, or Bayesian optimization will find a combination that makes the model very effective. Additionally, try different learning rate schedules and optimization algorithms that will further tune the training to optimize the performance of the model.

One of the newest ways to improve the capabilities of NER is through the investigation of multi-task learning frameworks. Multi-task learning simply means that the model should be trained on several tasks that are somehow connected, for example, training on NER and sentiment analysis or text classification. In this case, it would enable the model to learn richer representations of the text and grasp finer relationships among various tasks. For effective multi-task learning, there is a need to balance properly between the learning objectives for all tasks and ensure successful knowledge transfer. Imbalance in classes is a very critical issue for the overall NER task. Balancing the class distributions is vital for ensuring the required performance of the model against all types of entities. These can include oversampling underrepresented classes, undersampling overrepresented classes, or introducing class-weighted loss functions to minimize the effect of imbalance. Besides, generating synthetic data only for the underrepresented classes may result in a more balanced training set and allow improving the model's ability to recognize all entities more effectively.

The training of a BERT based NER model on the Enron Email dataset was quite promising in the capabilities presented by the model in the understanding and classification of the named entities, hence still illuminating BERT's effectiveness in carrying out this task in the face of data limitation challenges and class imbalance. To address these issues and push the boundary of this research even further, future work will need to attempt to increase and diversify datasets, use synthetic data augmentation, try more sophisticated hyperparameter tuning, and adopt the multi-task learning schema. These will make the proposed model perform better, supporting more flexible and powerful NLP applications in the future. This project demonstrates not only the potential of BERT in NER but also sets the way for further innovations in NLP technologies.

6 FUTURE ENHANCEMENTS:

This section serves to identify areas of improvement and future research directions. Specifically, here are several ways the overall outcome of the project could be improved:

Expanding the Dataset: Dataset play crucial role to increase the model's robustness and accuracy so it is essential to expand the dataset by including a more diverse range of email samples.This diversity can improve the model's ability to generalize across different contexts and email types, ultimately leading to more accurate information extraction. Also, a large data set with different examples will reduce overfitting-when a model does well on training data but then poorly on other data. Ultimately, it will lead to a robust dataset, and the performance in all information extraction tasks will be increased because the model is going to precisely track and correctly classify any entity in a real situation.

Improving Data Quality: The most important steps toward reliable and accurate results in any machine learning project is improving data quality. Future work should focus on improving data quality through thorough cleaning and consistent annotation. In doing so, one avoids ambiguity and thus enables models to learn from examples that are crystal clear. If the dataset is right and free of inconsistencies, then this will drive big strides in terms of performance by yielding more correct entity recognition and classification in real-world applications. Ultimately, that will translate to a more robust model with reliable insights from various email content.

Using Additional Approaches: Future researchers should investigate state-of-the-art NER models to enhance information extraction systems. This would be effectively done through the use of conditional random fields, which consider the dependencies between words that are neighbors to each other, hence recognizing entities more correctly, especially in the processing of emails where context is very important. More so, domain-specific knowledge can help to improve contextual understanding and increase accuracy in areas that are specifically specialized. Also to be regarded are ensemble modeling, a technique which provides better performance by aggregging predictions of several different models, and active learning, where the model itself decides how many of the

most informative samples should be labeled for improvements of the predictions. These techniques can greatly improve both performance and reliability in the processing of different emails by NER systems.

Recommendations for future researchers: The researchers should be very careful while selecting the most compatible NER model with the project and the set goals. This is a critical selection to have the model face specially existing challenges in this particular job. Besides, data preparation is one of the most important things. High-quality training data is a must in obtaining accurate and reliable results. This is also supported by the fact that when doing NER effectively, consistent labeling in the data is important to make the entity tagging very clear and uniform.To improvise entity recognition, the researchers can add another layer that could be a Conditional Random Fields, helping the model in recognizing entities by understanding the relationship among the nearby words. While training the model, it's essential to avoid overfitting, where the model learns the training data too well but performs poorly on new data. To prevent this, researchers should use techniques like early stopping, which halts training when performance on a validation set starts to decline, and regularization methods to control model complexity. Also, proper evaluation has to be taken out, including numbers and descriptions to find areas that need improvement.This is a process recommended to be undertaken with experts in the field, whose knowledge will add great context and consequently improve model accuracy. In such collaboration, researchers can tune these NER systems and thus make them much more attuned to their area of operation for effective information extraction.

7 CONCLUSION:

This project, entitled ”INFORMATION EXTRACTION FROM EMAIL USING BERT,” is aimed at developing an automated information extraction system based on the BERT model. Major objectives of this work were to extract relevant information from email content and attachments, especially of type .docx, and automate it to bring in efficiency and accuracy.

The system was successfully implemented, extracting key entities like the sender, item ID, quantity, unit of measure, cost, and order number from the email and its attachments. The BERT model showed high accuracy in identifying most entities, with perfect precision, recall, and F1-scores for the ”order number” and ”sender.”. It also showed a capability for cleaning and processing email content effectively so that, after extraction of the data, it is securely and accurately stored in a database.Model performance was less strong in certain areas. For example, on ”Cost,” the precision of the model was high, but recall was only modest, which could lead to under-identification of this entity. The ”Product Name” entity caused the most problems for precision, recall, and F1-scores among all entities, indicating that some works were needed to make the model more robust when facing changes in product names.

The objectives set at the beginning of the project were largely met. Useful information was extracted from the email messages and their .docx attachments, showing well how the BERT model is working in a real-life application. Besides, the Celery background tasks were implemented to fetch unseen emails and process them in the system, so it could continuously run and update data processing.Despite achieving high accuracy, a number of challenges and errors have been noted, which identify scope for improvements in the future. It was the project that laid the foundation for automated information extraction, applying a comprehensive approach to management and processing email data. Findings and insights attained through this project will be valuable guidance for research and development in this area in the future.

APPENDIX A

A.1 Proposal Schedule

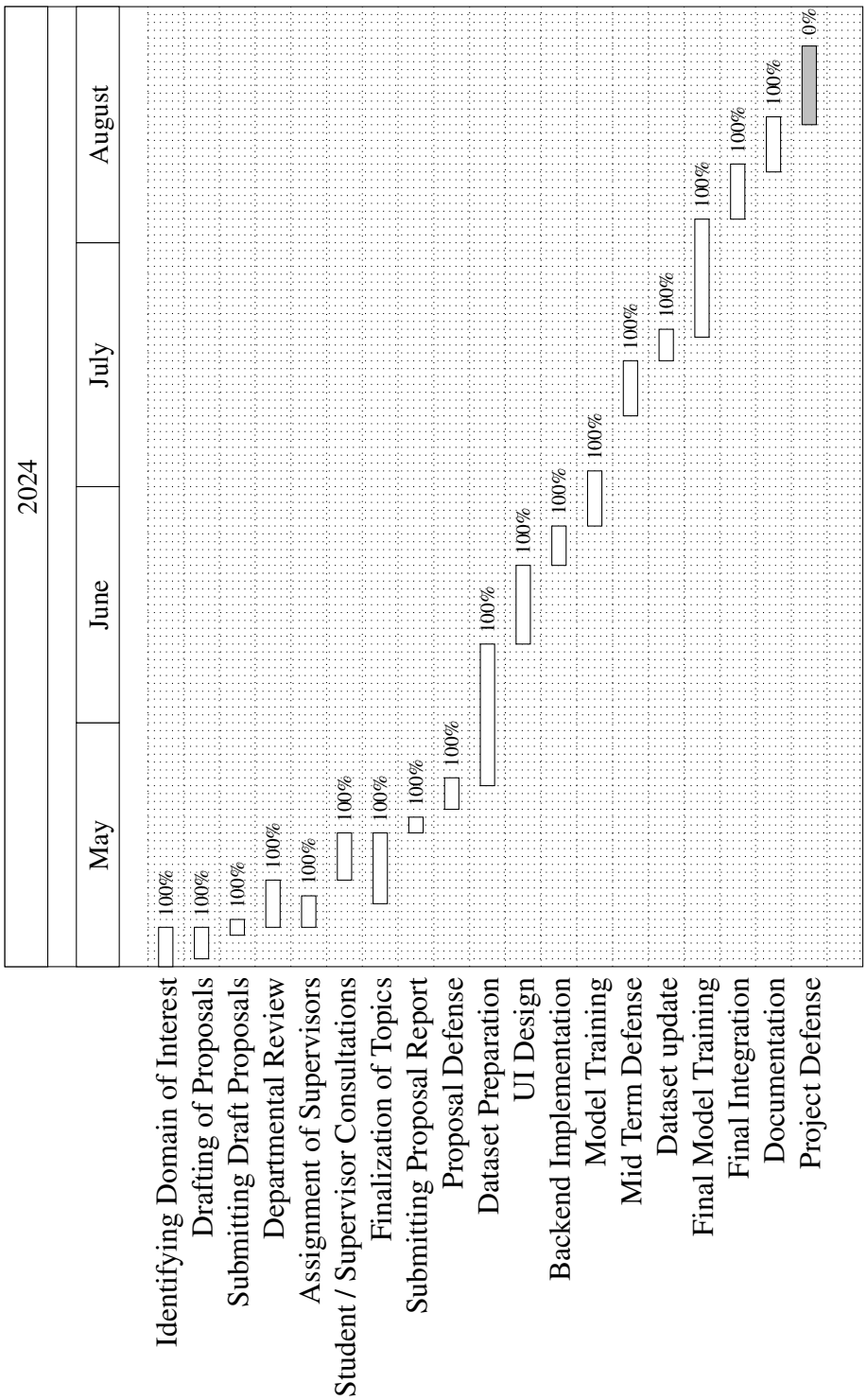

Figure A.1: Gantt Chart showing Project Timeline.

A.2 Literature Review of Base Paper- I

Author(s)/Source: Francis G. VanGessel , Efrem Perry, Salil Mohan, Oliver M. Barham, Mark Cavolowsky

Title: Natural Language Processing for Knowledge Discovery and Information Extraction from Energetics Corpora

Website: `https://onlinelibrary.wiley.com/doi/abs/10.1002/prep.202300109`

Publication Date: 06 Oct 2023	**Access Date: 03 May 2024**
Journal: Wiley Online Library	**Place:** n/a
Volume: n/a	**Article Number:** n/a

Author's position/theoretical position: n/a

Keywords: Detonation Science, NLP Knowledge Discovery Large Language Models

Important points, notes, quotations **Page No.**

1. The study employ unsupervised learning algorithms, including Latent Dirichlet Allocation (LDA), Word2Vec (W2V), and Transformers **08**
2. To visualize the relationships between these topics, t-SNE a dimensionality reduction technique used to project the high-dimensional data (300 dimensions) into a two-dimensional space. **12**
3. Fine-tuning Transformer model weights improves its understanding of the energetics domain. **15**
4. This improvement is likely due to the Transformer's attention mechanism, which captures context-dependent relationships between words in the abstract. **17**

Essential Background Information: Most recent studies use complex DL models trained on thousands of molecules to achieve high accuracy.

Overall argument or hypothesis: Training ML and DL models typically requires large datasets, which can be difficult to obtain in energetics due to safety and security concerns.

Conclusion: This study explored three NLP models on a large corpus of energetics text data. All three models captured domain-specific knowledge, with the Transformer model performing best. An NLPbased document classification pipeline achieved high accuracy 76% with Transformer.

Supporting Reasons

1. LDA (Latent Dirichlet Allocation) discovers hidden topics in text data without needing manual labeling of each document.

2. Word2Vec creates word embeddings that capture relationships between words.

3. Transformers excel at understanding long-range dependencies in sentences, crucial for comprehending the intricacies of scientific writing in energetics where connections between concepts spread across sentences matter.

8. Multiple ML models were trained to predict gait parameters and tested.

Strengths of the line of reasoning and supporting evidence: The research illustrates how NLP techniques can be effectively applied in energetics research, showcasing their viability and usefulness. This is evidenced by achieving classification accuracies of 74% and 76% using NLP algorithms for document categorization.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: The corpus size (80,000 energetic documents) is small compared to other LLMs training datasets.

A.3 Literature Review of Base Paper- II

Author(s)/Source: Mr. Chandrashekhar Mankar, Adarsh Mundada, Sahil Nagrale,Pavan Malviya, Aniket Sangle,Manoj Navrange	
Title: BART Model for Text Summarization : An Analytical Survey and Review	
Website: `https://www.researchgate.net/publication/360633194`	
Publication Date: May 2022	**Access Date: 03 May 2024**
Publisher or Journal: IJARSCT	**Place:** n/a
Volume: 2	**Issue Number:** 5
Author's position/theoretical position: Student at Shri Sant Gajanan Maharaj College of Engineering	
Keywords: BART [Bidirectional and Auto-Regressive Transformers], BART [Bidirectional Encoder Representations from Transformers] , GPT [Generative Pre-trained Transformer].	

Important points, notes, quotations **Page No.**

1. BART (Bidirectional and Auto-Regressive Transformers) model is utilized for text summarization, employing a denoising autoencoder approach for pretraining. **02**
2. Noise reduction strategies such as rearranging phrase sequences and employing text infilling with mask tokens have shown promising results in BART's performance. **04**
3. Remote disease classification on the largest cohort of participants. **134**
4. Self-supervised approaches like masked language models have been successful in NLP tasks, with BART presenting advancements in pretraining techniques. **05**

Essential Background Information: Two primary techniques in text summarization are extractive, which selects significant parts of the original text, and abstractive, which interprets and rephrases content in natural language.

Overall argument or hypothesis: BART's flexibility and effectiveness make it a valuable tool in handling large volumes of text data and extracting key information efficiently.

Conclusion: Clinical features derived from wearable sensors can perform disease classification and severity prediction on a diverse population.

Supporting Reasons

1. BART's denoising autoencoder approach allows for effective pretraining, demonstrating proficiency in both text creation and comprehension.

2. The model's transformer-based architecture combines elements from BERT and GPT

3. Experimentation with noise reduction strategies has shown significant improvements in BART's performance.

4. Comparative analysis with other models like RoBERTa and GPT showcases BART's state-of-the-art performance in various NLP tasks, including summarization

Strengths of the line of reasoning and supporting evidence: BART's effectiveness in text summarization, supported by clear methodology, thorough noise reduction strategy evaluation, and comparative analysis with existing models. Integration of theory with practical results and alignment with current research trends further strengthen the argument.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: It lacks depth in evaluating metrics and exploring model limitations. Long-term evaluation is also neglected, impacting understanding of BART's sustainability. Addressing these gaps is crucial for a more robust argument on BART's efficacy in summarization.

A.4 Literature Review of Base Paper- III

Author(s)/Source: Xinlu Li , Yuanyuan Lei and Shengwei Ji	
Title: BERT- and BiLSTM-Based Sentiment Analysis of Online Chinese Buzzwords	
Website: `https://doi.org/10.3390/fi14110332`	
Publication Date: June, 2017	**Access Date:** 03 May 2024
Publisher or Journal: n/a	**Place:** n/a
Volume: Future Internet 2022, 14	**Issue Number:** 332
Author's position/theoretical position: Master's Student	
Keywords: online Chinese buzzwords; sentiment analysis; deep learning; pre-trained language models; BiLSTM	

Important points, notes, quotations — **Page No.**

1. The model doesn't necessitate word separation for sentiment analysis and can grasp profound contextual details from the word order. **03**
2. BERT helps understand the contextual meaning of words in OCBs, addressing the challenge of unclear sentiment. **04**
3. BiLSTM analyzes the sequence of words, capturing shifts in sentiment within an OCB. **58**
4. Manual labeling and classification were conducted on the text content, with negative OCBs represented by 0 and positive OCBs represented by 1, ensuring data validity. **09**

Essential Background Information: Sentiment analysis methods include lexicon-based, machine learning, and deep learning. Lexicon-based relies on manual dictionaries like SentiWordNet . Machine learning uses algorithms such as Logistic Regression or SVMs on word frequency. Deep learning employs CNNs or RNNs for nuanced sentiment analysis.

Overall argument or hypothesis: A sentiment analysis model combining pre-trained BERT and BiLSTM can effectively classify the sentiment orientation (positive, negative, or neutral) of Online Buzzwords (OCBs) despite their informal structure and unclear emotional cues.

Conclusion: Proposed model achieved good results on a network catchphrase dataset. However, the authors acknowledge limitations in the data size, restricting deeper analysis of emotional features. They plan to expand the data scope in future research to improve the accuracy of sentiment analysis for internet language.

Supporting Reasons

1. Integrates BERT and BiLSTM for deep contextual understanding.

2. Effectively manages irregular OCB structures.

3. Superior recall and F1-score over state-of-the-art models.

4. BERT fine-tuning speeds up model convergence.

5. Avoids word separation, ideal for OCB sentiment analysis.

6. Captures local and global semantic features for better classification.

Strengths of the line of reasoning and supporting evidence: A promising approach to sentiment analysis for Online Buzzwords. By addressing the limitations in data size and providing more specific results, the researchers can further solidify their arguments and refine their model for even more accurate sentiment analysis of the ever-evolving language of social media.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: Dataset might not be large enough to capture a wider range of emotional features.

A.5 Literature Review of Base Paper- IV

Author(s)/Source: Tao Ni , Qing Wang	
Title: Explore BiLSTM-CRF-Based Models for Open Relation Extraction	
Website: https://www.researchgate.net/publication/351105323_Explore_BiLSTM-CRFBased_Models_for_Open_Relation_Extraction	
Publication Date: Nov 21 2021	**Access Date:** 03 May 2024
Publisher or Journal: Research gate	**Place:** School of Graduate Studies
Volume: n/a	**Issue Number:** n/a
Author's position/theoretical position: Australian National University, Australia	
Keywords: Bidirectional LSTM-CRF , Contextualized Word Embeddings , Multiple Relation Extraction , Tagging Scheme	

Important points, notes, quotations — **Page No.**

1. Tagging scheme for Open Relation Extraction reduces overlapping issues, enhancing model performance. **02**
2. BiLSTM-CRF models for Open RE tasks, utilizing BiLSTM networks for sequence analysis and CRF layers for sequence tagging **03**
3. BERT-based BiLSTM-CRF model shows superior performance, leveraging contextual word embeddings for more accurate relation extraction, especially in sentences with multiple relations **06**
4. New tagging scheme combined with BiLSTM-CRF models leads to significant improvement in performance metrics like recall and PMS, indicating better relation extraction capability **05**

Essential Background Information: : Sequence Tagging Models use recurrent neural networks (RNNs) to label each word in a sentence and extract relations and their arguments. Sequence-toSequence Models view Open RE as a machine translation task.

Overall argument or hypothesis: Existing Open RE models struggle with sentences containing multiple relations. BiLSTM-CRF network with different word embedding methods to achieve better performance.These model utilizes both pre-trained word embeddings (GloVe, Word2Vec) and contextualized word embeddings (BERT, ELMo) to capture word meaning and context.

Conclusion: The new tagging scheme reduces overlapping issues and improves the model's ability to identify relations.The BERT-BiLSTM-CRF model achieves the best overall performance, correctly extracting over 94% of relations in sentences with multiple relations.

Supporting Reasons

1. New tagging scheme resolves overlapping issues.

2. Advanced word embeddings enhance model performance.

3. Comprehensive experiments ensure robust evaluation.

4. Introduction of Predicate Matching Score (PMS) metric.

5. Comparison with existing methods validates effectiveness.

6. Future research directions offer promising avenues for exploration.

Strengths of the line of reasoning and supporting evidence: The challenge of extracting multiple relations in Open RE by introducing an innovative tagging scheme, utilizing advanced word embeddings, and conducting comprehensive experiments.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: The paper focuses on BiLSTM-CRF models. While it compares different word embedding methods (GloVe, Word2Vec, BERT, ELMo), it would be stronger if it compared the BiLSTM-CRF architecture to other relation extraction architectures like Transformers .

A.6 Literature Review of Base Paper- V

Author(s)/Source: Shehab Abdel-Salam , Ahmed Rafea	
Title: Performance Study on Extractive Text Summarization Using BERT Models	
Website: https://www.mdpi.com/2078-2489/13/2/67	
Publication Date: 28 January 2022	**Access Date: May, 2024**
Publisher or Journal: MDPI	**Place:** , The American University in Cairo
Volume: 13	**Issue Number:** n/a
Author's position/theoretical position: Master's Student	
Keywords: Computer vision; Deep learning; Disease management; Health monitoring; Parkinson's disease	

Important points, notes, quotations — **Page No.**

1. Involves three phases: pre-processing, algorithmic processing, and post-processing. **2**
2. DistilBERT reduces BERT model size by 40% while maintaining 97% of its language understanding capabilities. **05**
3. SqueezeBERT retains 98% of baseline performance with half the size of BERT-base. **05**
4. DistilBERT scores slightly lower than BERT-base in ROUGE metrics. **04**
5. SqueezeBERT can be deployed for real-time summarization. **08**

Essential Background Information: The effectiveness of DistilBERT and SqueezeBERT, compressed versions of BERT, for extractive text summarization, highlighting their potential for real-time summarization tasks due to reduced model size while maintaining performance levels compared to the standard BERT model.

Overall argument or hypothesis: Models like DistilBERT and SqueezeBERT can summarize text effectively like BERT but faster and with less complexity, making them suitable for quick summarization tasks.

Conclusion: The results show that DistilBERT and SqueezeBERT exhibit competitive performance in extractive text summarization while offering reduced model sizes, suggesting their suitability for real-time summarization tasks.

Supporting Reasons

1. Limited dataset diversity may restrict the study's applicability across various domains.

2. The trade-offs between model size reduction and summarization quality .

3. Focuses on extractive summarization, leaving out potential insights into abstractive summarization.

4. SqueezeBERT, utilizing grouped convolutional layers for efficiency,

5. SqueezeBERT achieves similar or slightly better ROUGE scores compared to DistilBERT.

6. Despite the reduction in parameters, DistilBERT achieves competitive ROUGE scores compared to the BERT-base model.

Strengths of the line of reasoning and supporting evidence: Effectiveness in text summarization through rigorous experimentation and comparison with baseline BERT models.Smaller models are suitable for real-time summarization tasks, highlighting their strengths in efficiency and performance.

Flaws in the argument and gaps or other weaknesses in the argument and supporting evidence: Dataset variety may affect its applicability, and it could delve deeper into potential trade-offs between model size reduction and summarization quality to provide a more nuanced understanding of DistilBERT and SqueezeBERT's performance.

REFERENCES

[1] Mario Franco Sarto, Lorenzo Morselli, Alessia Campi, and Fausto Giunchiglia. Knowledge-based management of patients in intensive care units: An ontology-driven approach. *Information*, 13(2):67, 2022.

[2] X. Li, Y. Lei, and S. Ji. Bert- and bilstm-based sentiment analysis of online chinese buzzwords. *Future Internet*, 14(11):332, 2022.

[3] C. Mankar et al. Bart model for text summarization: An analytical survey and review. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, 2(5), may 2022.

[4] T. Ni, Q. Wang, and G. Ferraro. Explore bilstm-crf-based models for open relation extraction. `https://www.researchgate.net/publication/XXXXXXX`, apr 2021.

[5] Kayal Padmanandam, KVN Sunitha, Behafarid Mohammad Jafari, Ali Jafari, Mengyuan Zhao, and Nikitha Pitla. Customized named entity recognition using bert for the social learning management system platform coursenetworking. *Journal of Computer Science*, 20(1):88–95, 2023.

[6] F. G. VanGessel, E. Perry, S. Mohan, O. M. Barham, and M. Cavolowsky. Natural language processing for knowledge discovery and information extraction from energetics corpora. *Prep*, oct 2023. First published: 06 October 2023.