**EXTRACTING EVENT INFORMATION FROM**

**GROUP CHAT CONVERSATIONS**

**A DISSERTATION SUBMITTED TO**

**SVKM’S NMIMS (DEEMED TO BE UNIVERSITY)**

**IN PARTIAL FULFILMENT FOR THE DEGREE OF**

**MASTERS OF SCIENCE**

**IN**

**STATISTICS AND DATA SCIENCE**

**BY**

**FARIDA ATTAR**

**AKSHAT CHHEDA**

**SURAJ GIRAMKAR**

**PARISI JARIWALA**

**MADHUMAYI PARAKALA**

**AASTHA SHARMA**

**UNDER THE SUPERVISION OF**

**Dr. Pradnya Khandeparkar**

**NILKAMAL SCHOOL OF MATHEMATICS, APPLIED STATISTICS AND ANALYTICS**

**SVKM’s Narsee Monjee Institute of Management Studies**

**(Deemed-To-Be-University)**

**V.L. Mehta Rd, Vile Parle (West), Mumbai – 400056**

**April- 2024**

**GROUP INFORMATION:**

(GROUP 11)

| NAME | SAPID | PROGRAM | YEAR |
| --- | --- | --- | --- |
| Farida Attar | 86062300060 | MSC STATISTICS AND DATA SCIENCE | 2024-25 |
| Akshat Chheda | 86062300016 | MSC STATISTICS AND DATA SCIENCE | 2024-25 |
| Suraj Giramkar | 86062300052 | MSC STATISTICS AND DATA SCIENCE | 2024-25 |
| Parisi Jariwala | 86062300057 | MSC STATISTICS AND DATA SCIENCE | 2024-25 |
| Madhumayi Parakala | 86062300044 | MSC STATISTICS AND DATA SCIENCE | 2024-25 |
| Aastha Sharma | 86062300064 | MSC STATISTICS AND DATA SCIENCE | 2024-25 |

**ACKNOWLEDGEMENT**

We would like to extend our heartfelt gratitude to the Programme Chairperson of M.Sc. Statistics and Data Science, Prof.Dr. Pradnya Khandeparker, for her unwavering support and guidance during our project, "Extracting Event Information from Group Chat Conversations". Her encouragement and invaluable feedback have been pivotal in our successful completion of this endeavour.

Our sincere appreciation also goes to Mr.Rutik Bhurke, whose expertise in Machine Learning has greatly enriched our project journey. His insightful guidance has provided us with invaluable knowledge and skills.

We would like to express our gratitude to all faculty members, friends, and family who have supported us throughout this endeavour. Their encouragement and motivation have been a constant source of inspiration.

**ABSTRACT**

Group chats have emerged as indispensable tools for communication within educational institutions, offering both asynchronous and instant messaging capabilities along with the convenience of sharing links and content. However, the pervasive nature of these chats poses challenges for users in managing the overwhelming volume of conversational content and staying updated after periods of offline activity. Additionally, the lack of structured organisation within chat feeds hinders efficient management of interruptions and turn-taking, making it difficult for users to focus on relevant information, particularly when attempting to catch up on missed content. In response to these challenges, this study proposes a solution to extract pertinent information such as schedule changes and event announcements from chat conversations and integrate it into an event calendar. The objective is to develop a database capable of extracting and cataloguing key events, along with their associated dates, to provide users with a more streamlined and accessible means of accessing critical information. Through this approach, the aim is to alleviate information overload within group chats and enhance the effectiveness of communication within educational settings.

**CONTENTS**

1. Introduction
2. Objective
3. Literature Review
4. Data Description
5. Data Cleaning
6. Methodology
7. Application
8. Results
9. Final Output
10. Conclusions
11. Limitations
12. Future Scope
13. Bibliography

**INTRODUCTION**

Group chats have emerged as a ubiquitous tool for collaboration, particularly within educational institutions where platforms like WhatsApp play a crucial role. These communication channels facilitate instantaneous, asynchronous interaction among individuals spanning various geographic locations and time zones, allowing for the seamless sharing of links and content. They also serve as a means to promptly disseminate updates on forthcoming events or alterations in schedules.

However, the perpetual nature of group chats, characterized by a constant influx of conversational content, presents challenges for users in managing the volume of information and catching up after periods of offline activity. Moreover, the structure of chat feeds often leads to difficulties in managing interruptions and navigating through interleaved discussions on diverse topics. Consequently, users returning to the conversation after an absence, particularly of considerable duration, may struggle to focus their attention and identify pertinent issues or events, risking overlooking crucial dates and updates.

To address these challenges, there is a need for an organized system capable of extracting and presenting missed content in a coherent manner. Existing information extraction systems for short text typically generate uniform summaries, while prior research in this domain has predominantly focused on extracting information from textual documents rather than chat messages. Furthermore, many of these methods necessitate explicit user input such as interest aspects, queries, or additional metadata like tags, comments, or ratings.

In response to these limitations, we propose a group chat information extraction system. Leveraging attributes such as participant identity, timestamp, and location, our model identifies and prioritizes the most relevant conversations within the chat. The extracted information is then presented in a user-friendly format, enabling individuals to swiftly grasp the content they missed. Our overarching objective is to facilitate the extraction and annotation of user-relevant concepts from chat conversations, including the identification and extraction of any events mentioned therein.

**OBJECTIVES**

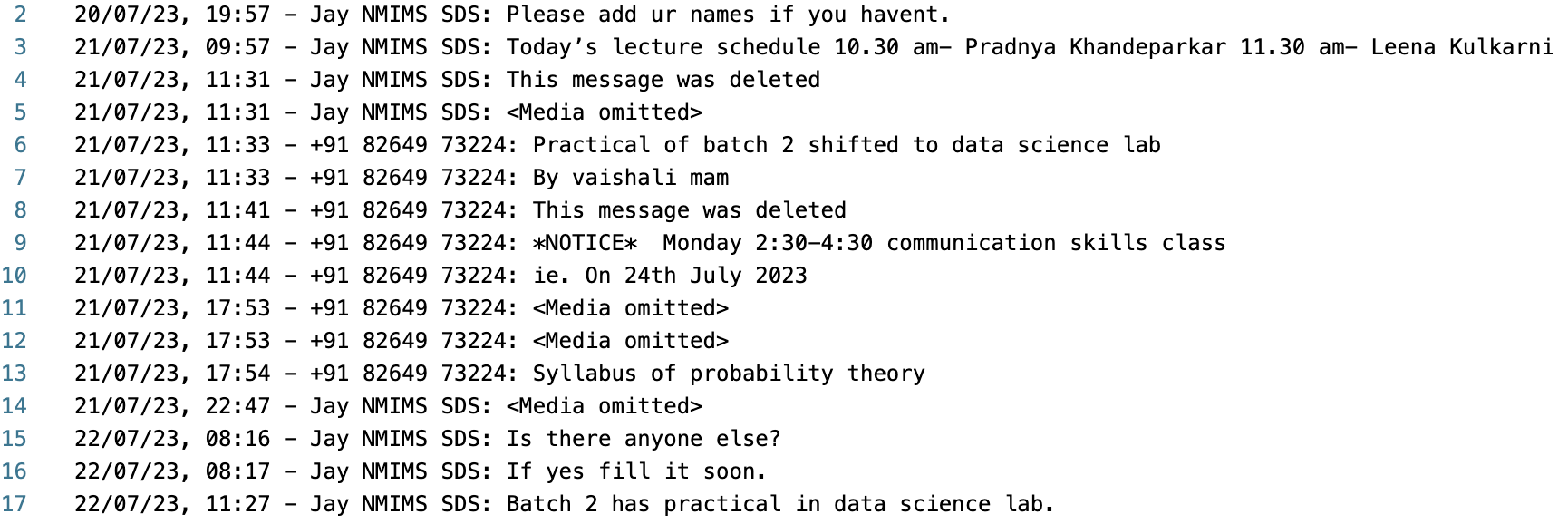
* Develop a system to extract personal information (such as names, locations, organisations, dates, and events) from group chat conversations
* Create a dynamic database infrastructure capable of efficiently storing and managing event-related data extracted from chat conversations.
* The database should include features for cataloguing events along with their respective dates, enabling users to easily search and retrieve event-related information.

**LITERATURE REVIEW**

* Tepper et al., worked on Collabot which makes system to learn about users interest and connections within a group chat without any direct information provided by the user.This allows collabot to generate summaries and recommendations for each user personally according to their needs and interest.
* Gupta et al,. introduce CONTEXT-NER, a task that aims to generate the relevant context for entities in a sentence, where the context is a phrase describing the entity but not necessarily present in the sentence.
* Huang et al., applies a bidirectional LSTM CRF (denoted as BI-LSTM-CRF) model to NLP benchmark sequence tagging data sets. They show that the BI-LSTM-CRF model can efficiently use both past and future input features thanks to a bidirectional LSTM component. It can also use sentence level tag information thanks to a CRF layer.
* Strubell et al., propose a faster alternative to Bi-LSTMs for NER: Iterated Dilated Convolutional Neural Networks (ID-CNNs), which have better capacity than traditional CNNs for large context and structured prediction.
* Akbik et al., present FLAIR, an NLP framework designed to facilitate training and distribution of state-of-the-art sequence labeling, text classification and language models. The core idea of the framework is to present a simple, unified interface for conceptually very different types of word and document embeddings. This effectively hides all embedding-specific engineering complexity and allows researchers to “mix and match” various embeddings with little effort.

**DATA DESCRIPTION**

The dataset comprises official group chat conversations from MSc SDS, ASA, and DS. These chats were extracted from WhatsApp and encompass text messages spanning from the inception of each chat group i.e in July 2023 to April 5th, 2024. The figure below is a snapshot of the dataset.

****

**DATA PREPROCESSING**

This section outlines the necessary functions and steps required for preprocessing the chat data obtained from WhatsApp, ensuring that it is suitable for further analysis and processing.

**1. DEFINE THE REQUIRED FUNCTIONS FOR PREPROCESSING:**

* Stop Word Removal (using nltk.corpus): Stop words are commonly used words that do not carry significant meaning and are often removed to improve text processing efficiency. We will utilise the nltk.corpus module to remove stop words from the text data.
* Checking for Typos (using TextBlob): TextBlob is a powerful Python library for processing textual data. We will employ TextBlob to check for typographical errors in the text, helping to ensure data accuracy and consistency.

**2. STEPS TO CONVERT TEXT FILE TO REQUIRED DATAFRAME:**

* Import Chats from WhatsApp into a .txt File: We will begin by exporting the chat conversations from WhatsApp into a text file format, preserving the original content.
* Rearrange Sentences in the Text File: To facilitate data processing, we will rearrange the sentences in the text file so that each new sentence starts where there is a date. This will involve writing Python code to automate the rearrangement process.
* Convert the .txt File to .csv File: The next step involves converting the text file into a CSV (Comma-Separated Values) format. The resulting CSV file will consist of a single column containing the text data.

**3. SPLIT THE COLUMN TO MULTIPLE COLUMNS USING DELIMITERS:**

* First Comma
* First Hyphen
* Second Colon (subject to the dataframe generated by the WhatsApp chats)

**4. CLEAN THE DATAFRAME:**

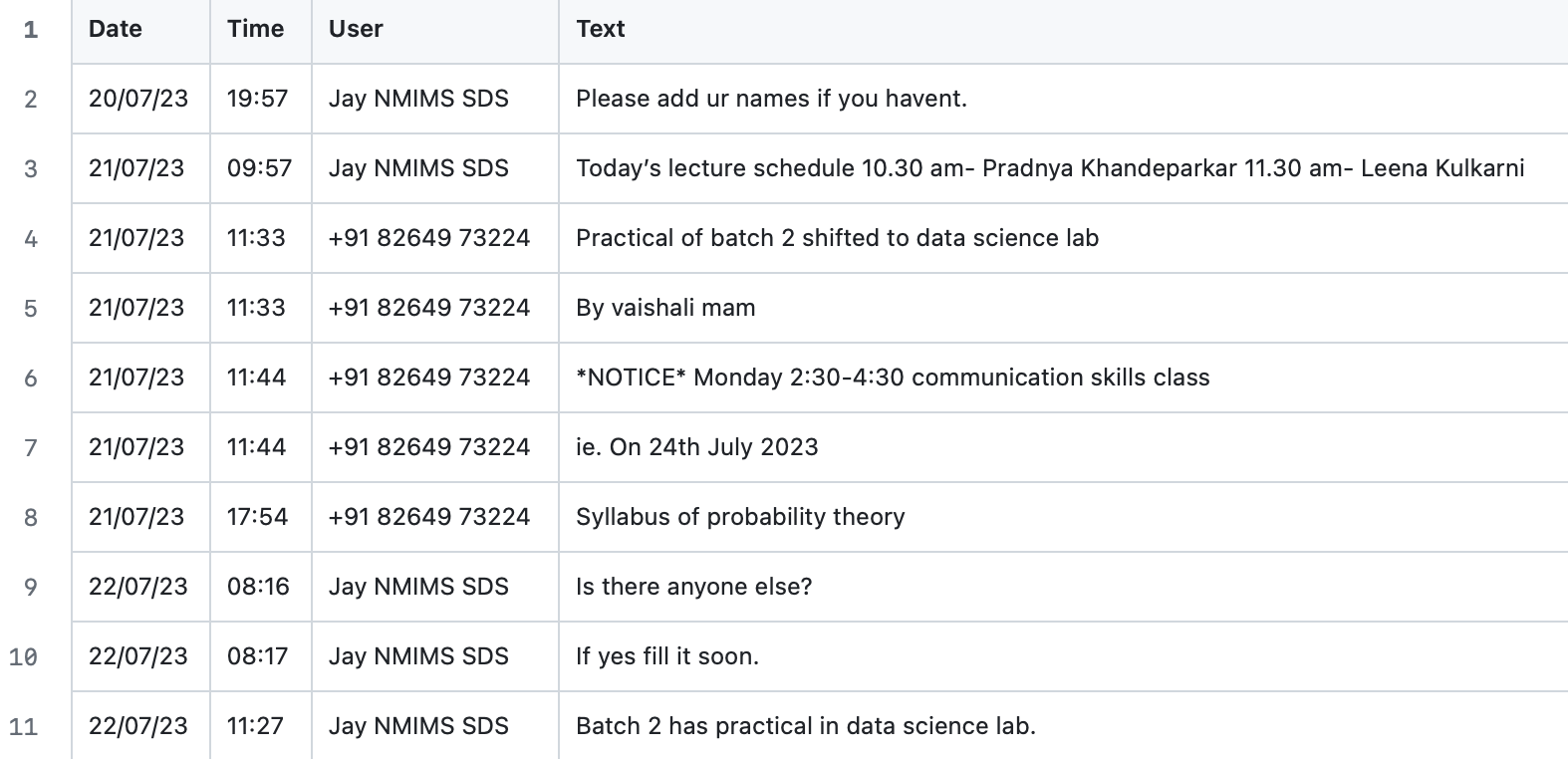
* Drop Null Values: Remove rows with missing 'user' or 'text' values.
* Remove Strings Containing "https://": Eliminate URLs present within the text data.
* Remove "This Message was Deleted": Exclude messages indicating deletion.
* Remove "Media Omitted": Discard messages containing omitted media.
* Remove Emojis: Eliminate emoji symbols from the text data.
* Convert all text to lowercase for seamless processing.

**5. PREPROCESSING CLEANED DATAFRAME:**

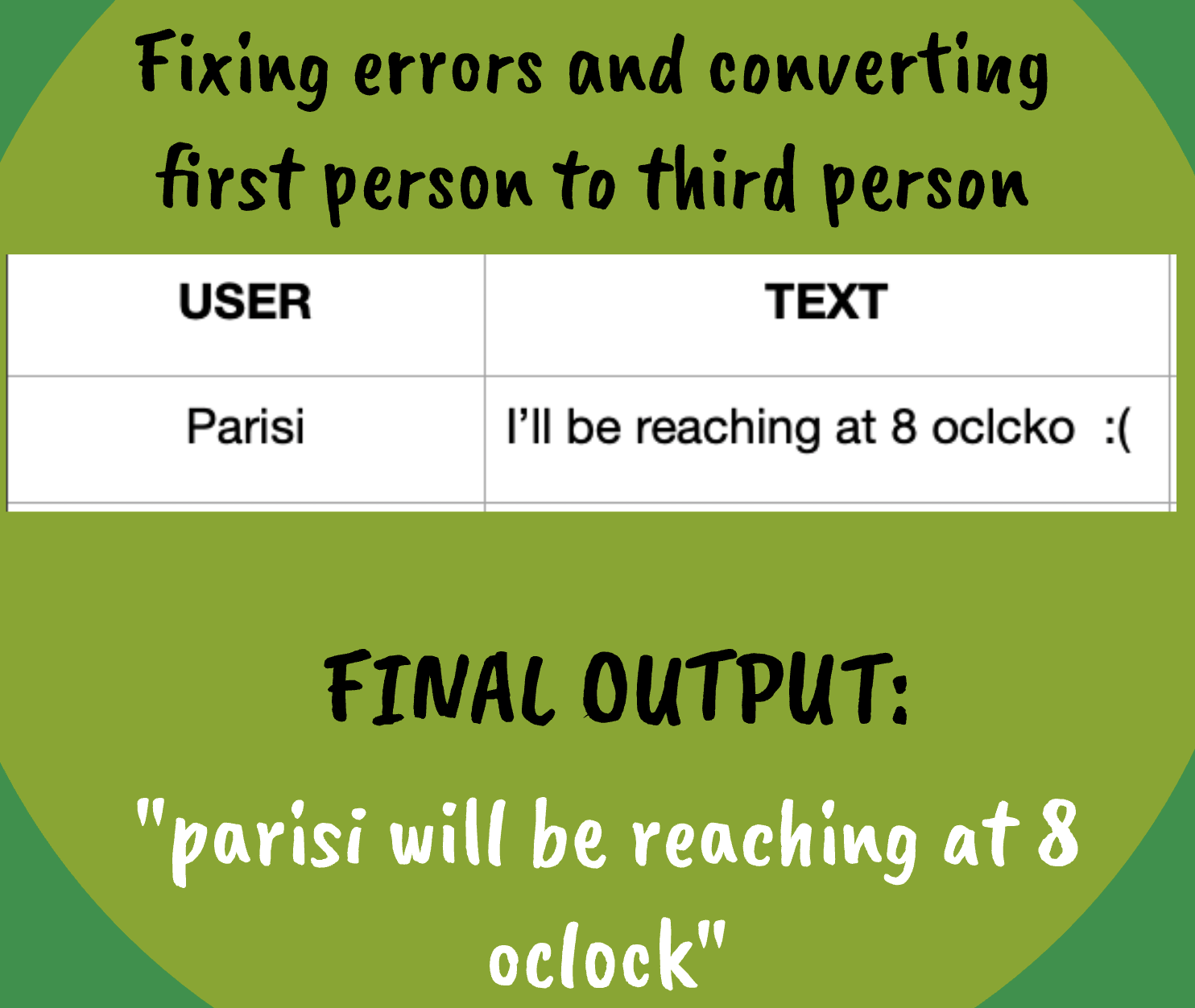
This step involves replacing specific text patterns with their standardized equivalents to ensure uniformity throughout the dataset. We'll address common linguistic contractions and expansions to enhance the clarity and coherence of the text data. For example,

* Replace "(I am)" with "(I is)"
* Replace "('ve)" with "(have)"
* Replace "('ll)" with "(will)"
* Replace "('m)" with "(am)"
* Replace "('won't)" with "(will not)"
* Replace "(n't)" with "(not)"

The figure below showcases the cleaned data frame.

****

In our ongoing efforts to refine the cleaned DataFrame for enhanced text consistency and readability, we've undertaken the transformation of text originally articulated in the first person to the third person perspective. This strategic alteration entails shifting the narrative viewpoint from the personal "I" or "we" to the more objective "he/she" or "they," thereby imbuing the text with a sense of detachment and impartiality. For example, take a look at the figure below

****

**METHODOLOGY**

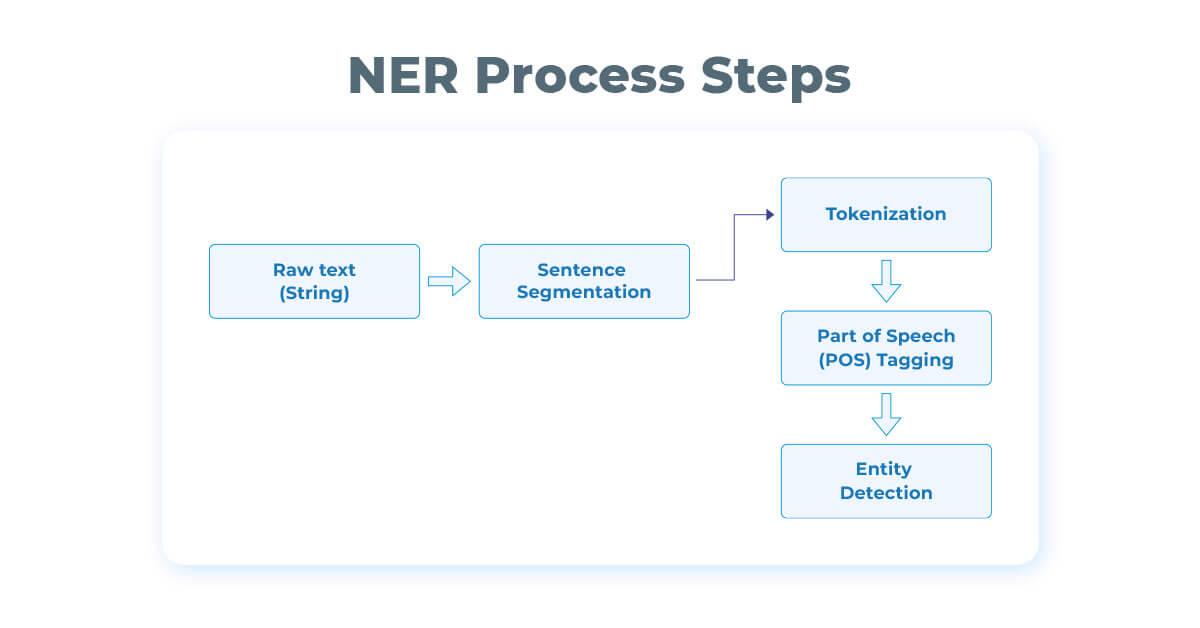
**NER**

NER stands for Named Entity Recognition. It's a natural language processing (NLP) technique used to identify and classify named entities within a text into predefined categories such as names of persons, organizations, locations, dates, numerical expressions, etc.

NER typically involves a sequence labelling task where each word or token in a text is classified into one of several categories. This is often done using machine learning algorithms, particularly deep learning models like recurrent neural networks (RNNs), long short-term memory networks (LSTMs), or transformer-based architectures like BERT.

The process of NER involves several steps:

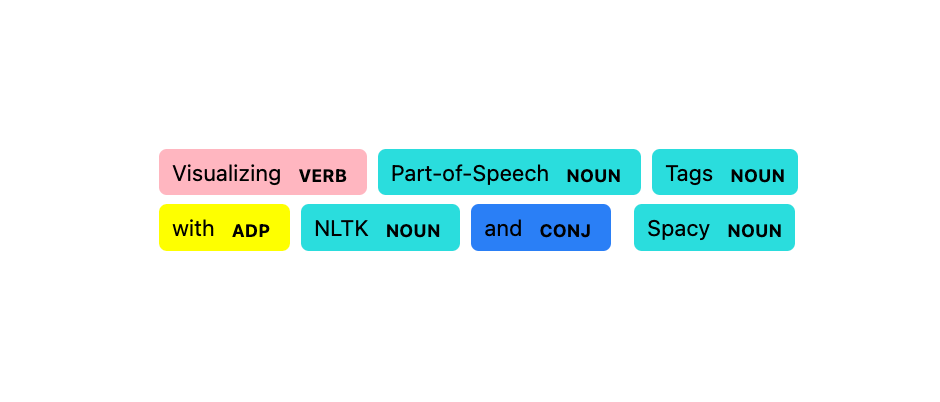
1. Tokenization: The text is split into individual words or tokens.
2. Feature Extraction: Features such as part-of-speech tags, word embeddings, or context windows are extracted for each token.
3. Classification: A classification model is applied to each token to determine its entity type.
4. Post-processing: Depending on the requirements, post-processing steps might be applied to improve the accuracy of the NER system. For example, rules might be applied to ensure that consecutive tokens with the same entity type are merged into a single entity.

****

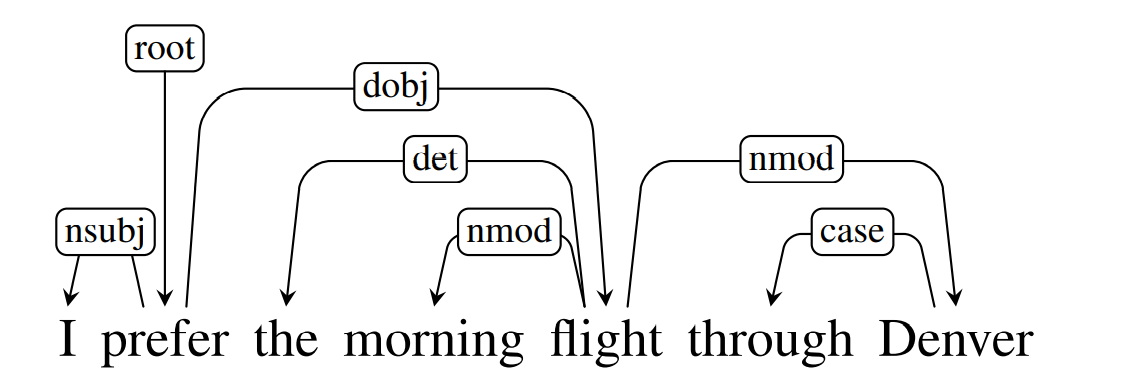
**spaCy**

spaCy is an open-source natural language processing (NLP) library designed for efficient and fast processing of natural language text. It provides a wide range of functionalities for tasks such as tokenization, part-of-speech tagging, named entity recognition, dependency parsing, and more. Here's how spaCy works:

1. **Part-of-Speech Tagging (POS)**: After tokenization, spaCy assigns each token a part-of-speech tag, indicating its grammatical category, such as noun, verb, adjective, etc. This process helps in understanding the syntactic structure of the text and is crucial for many downstream NLP tasks. Refer to the figure below.

****

1. **Dependency Parsing**: Dependency parsing involves analyzing the grammatical structure of a sentence by determining the relationships between words. spaCy uses dependency parsing to build a parse tree, where each word is a node, and the relationships between words are represented as directed edges. This information is useful for tasks like information extraction, question answering, and sentiment analysis.

****

1. **Named Entity Recognition (NER)**: Named entities such as persons, organizations, locations, dates, etc., are identified and classified within the text. This step is crucial for extracting structured information from unstructured text.

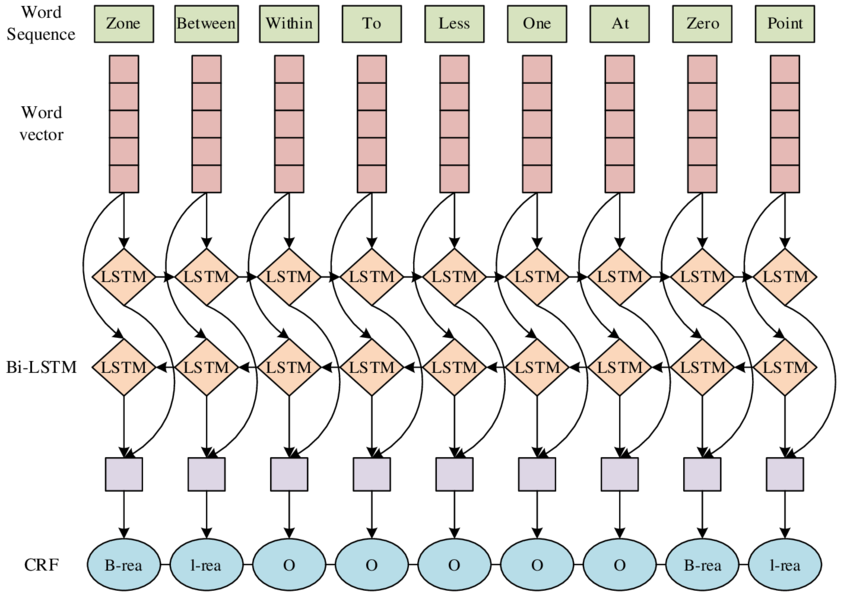
****

**FLAIR**

Flair is a natural language processing (NLP) library developed by Zalando Research for state-of-the-art text processing tasks. Flair provides a wide range of functionalities for tasks such as text classification, named entity recognition (NER), part-of-speech tagging (POS), sentiment analysis, and more. It utilizes a combination of bidirectional LSTM (Long Short-Term Memory) networks and Conditional Random Fields (CRFs) for named entity recognition (NER). Here's how it works

1. LSTM (Long Short-Term Memory): The LSTM component is responsible for capturing contextual information from the input text. It processes the input sequence token by token, maintaining an internal state that represents the context of each token based on preceding tokens. LSTMs are well-suited for capturing long-range dependencies in sequential data like natural language text.
2. CRF (Conditional Random Field): The CRF layer is used on top of the LSTM output to model dependencies between labels in the output sequence. CRFs are probabilistic graphical models that model the conditional probability distribution of label sequences given the input sequence. They take into account label dependencies and ensure globally consistent predictions by considering the entire sequence.

The combination of LSTM and CRF is a common approach for sequence labeling tasks like NER and POS tagging because it allows the model to capture both local contextual information (through LSTMs) and global label dependencies (through CRFs). Flair leverages this architecture to achieve state-of-the-art performance on such tasks. Refer to the figure below



**APPLICATION**

Before delving into the extraction of entities from the preprocessed data, it is imperative to identify the key attributes crucial for the project. Among these attributes, the most significant ones are time, person, and event, i.e., the when, who and what of the text messages. Let us proceed accordingly.

**WHEN:**

Identifying when the event will be conducted is very imperative for the project. This process involves a series of steps aimed at accurately predicting and standardizing time and date references within the text data:

1. **Using Spacy's Large Model for Prediction:** Spacy, a natural language processing library, is employed to predict time and date references within the text using its large pre-trained model.
2. **Utilizing Regular Expressions (RegEx) for Rule Definition:** RegEx patterns are defined to establish specific rules for text manipulation, such as:
   * Adding a space before "am" and "pm" to ensure proper recognition by Spacy's model.
   * Adding a space before "o'clock" if missing, enhancing pattern recognition by the model.
   * Converting time references to the 24-hour format for standardization.
3. **Extracting Time Indicators:** Special attention is given to extracting time references indicated by words such as "by," "at," or "in," which provide context for the timing of events or activities mentioned in the text.
4. **Recognizing Month Abbreviations:** Month abbreviations like 'Jan' or 'Feb' are identified within the text, and these references are extracted and categorized as temporal indicators denoting the 'WHEN' of events or occurrences.
5. **Using the Datetime Library for Conversion:** The datetime library is leveraged to convert temporal expressions like 'Today' and 'Tomorrow' into actual dates, accounting for the timing of the message transmission. This ensures that temporal references are accurately contextualised within the timeline of the conversation.

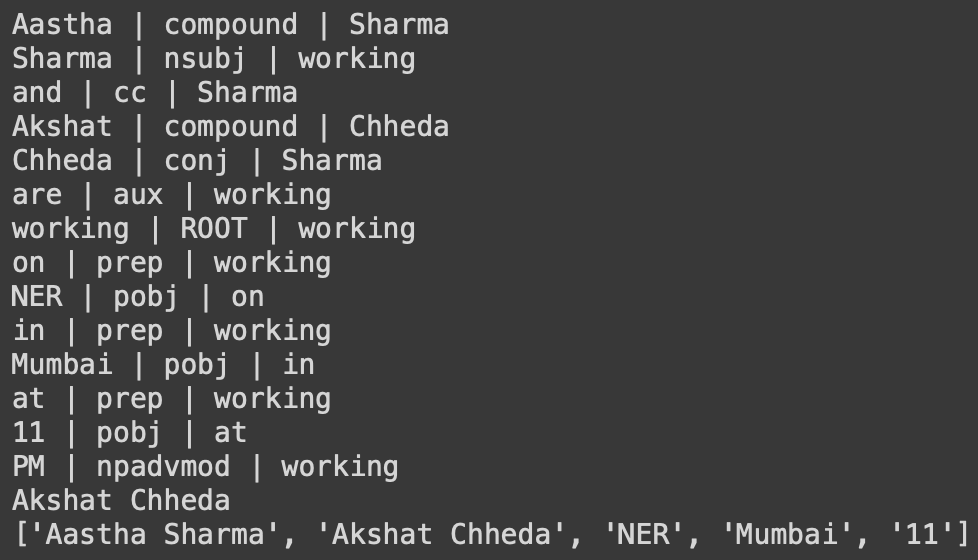
**WHO and WHAT:**

**Spacy’s Parser Tree**

Who and What will give us the subject and the context of the text messages. In order to determine the “who” part of the sentence, we’ve attempted to use Spacy’s Dependency Parser Tree. The output is shown in the figure below.

Suppose the sentence is

“Aastha Sharma and Akshat Chheda are working on NER in Mumbai at 11 PM.”

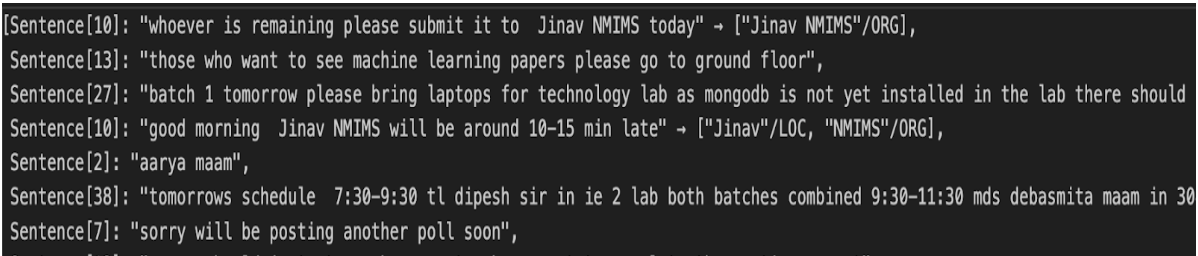
****

Through relation tags like “compound” and “conj”, the model can correctly determine Aastha Sharma and Akshat Chheda as “Persons”.

On similar lines, one can apply rules for each of the sentence types for the “WHO” column. However, manually creating rules for each sentence type in the "WHO" column may lead to overfitting due to the numerous grammatical sentence structures present. This underscores the need for a more generalized approach to accurately capture the subject of the text messages.

**Spacy Pre-trained Model:**

The adoption of a Spacy pre-trained model for identifying entities representing "Who" and "What" within text data marks a significant advancement in natural language processing (NLP) methodologies. Leveraging Named Entity Recognition (NER) tags pre-assigned to entities such as 'PER' for persons and 'ORG' for organizations, the model aims to automate the extraction of relevant information, circumventing the need for manual rule-based approaches. Refer to the example below

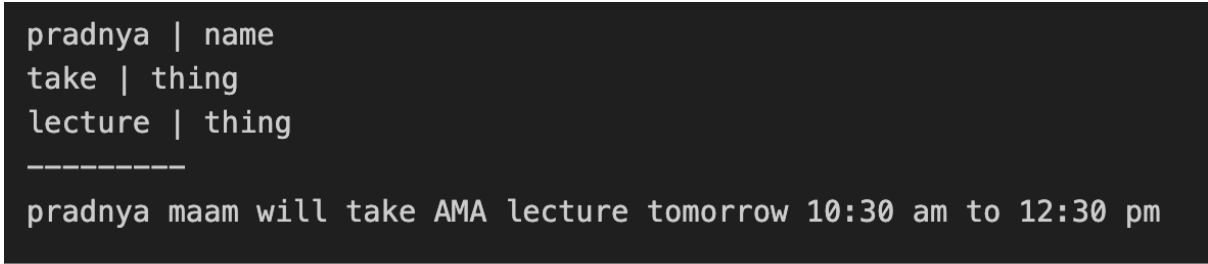


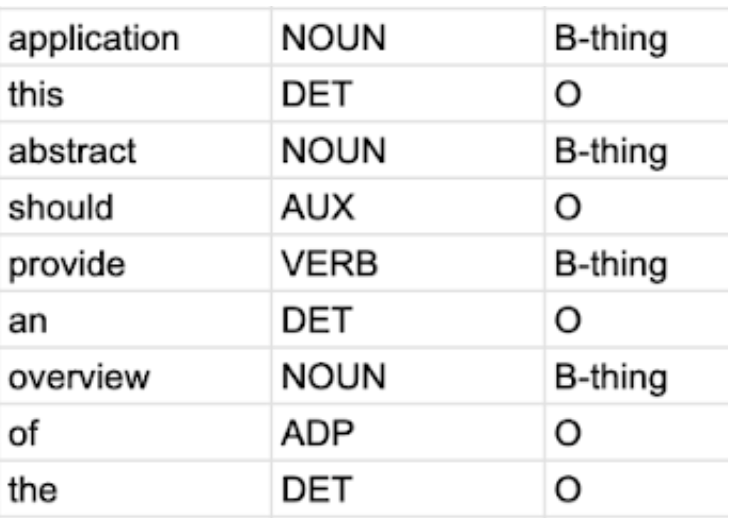
Using a pre-trained model for this task is generally considered more efficient and accurate compared to manual rule-based approaches. However, despite utilizing this advanced method, the results obtained did not meet the desired requirements. This suggests that while the Spacy pre-trained model provides a solid foundation, it may not fully capture the nuances or specific requirements of the task at hand, leading to suboptimal results.

**Spacy Pipeline:**

This process involves training a custom NER (Named Entity Recognition) model using the Spacy framework:

1. Spacy-ing the data: This step involves applying Spacy's linguistic processing pipeline to the input data. Each word in each sentence is analyzed using Spacy's tokenization, dependency parsing, part-of-speech tagging, and syntactic parsing functionalities. The output is organized into a structured format, typically a dataframe, for further processing.
2. Spacing the data: In this step, the data is formatted according to Spacy's requirements for training its language model (LLM). Each sentence is separated by a blank row, as mandated by Spacy's training conventions.
3. Annotating the Data: Manual annotation of the data is performed according to the specific requirements of the project. Entities relevant to "WHO" are annotated as "Name," while entities pertinent to "WHAT" are labeled as "Thing." Other entities are typically annotated as "O" (outside), indicating that they are not relevant to the targeted entities.
4. Train-Test-Validation Split: The annotated data is divided into separate sets for training, testing, and validation. This ensures that the model's performance can be evaluated on unseen data and helps prevent overfitting.
5. Converting the data to conll files: The data is converted from its original format (e.g., CSV) to the CoNLL format, which is a widely used standard for representing annotated textual data. Additionally, headers are typically removed from these files to conform to Spacy's requirements.
6. Creating the Spacy Pipeline: The processed and annotated data in the CoNLL format is used to train the custom NER model. Spacy's pipeline is configured with the training data, and the model is trained to predict NER tags based on the input text. The trained model is then evaluated on the validation set to assess its performance.





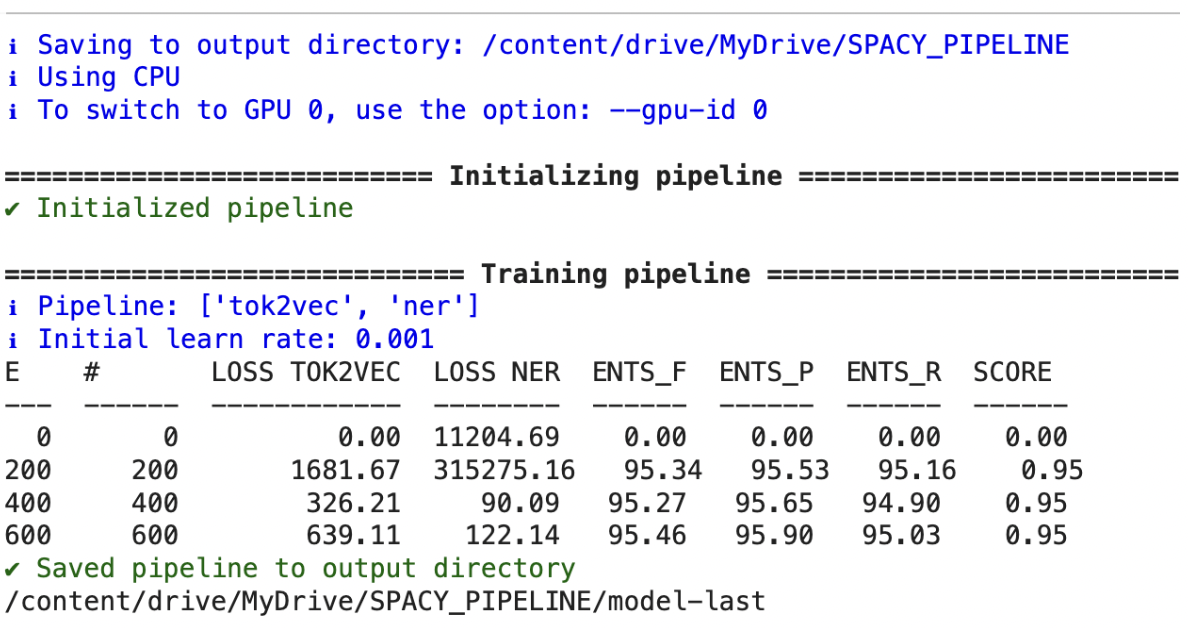
**Flair Model (iob):**

Alternatively, a Flair model is utilized for Named Entity Recognition (NER), specifically using the IOB (Inside, Outside, Beginning) tagging scheme. The IOB tagging scheme assigns labels to tokens in a sequence to indicate whether they are part of an entity and, if so, whether they represent the beginning, inside, or outside of the entity. For instance, a token at the beginning of an entity is labeled with "B-" followed by the entity type (e.g., "B-name" for the beginning of a person's name), while tokens inside the entity are labeled with "I-" followed by the entity type (e.g., "I-name" for words inside a person's name). Tokens that are not part of any entity are labeled as "O" (Outside).  
The annotations generated by the Spacy model are updated according to the IOB tagging requirements. This involves assigning appropriate IOB tags to tokens based on their positions within entities. Tokens at the beginning of entities are marked with "B-" tags, tokens inside entities are marked with "I-" tags, and all other tokens are marked with "O" tags.  
Once the annotations are updated to adhere to the IOB scheme, the data is processed further using the Flair model for NER training. The model is trained to predict NER tags based on the input text, with the IOB-tagged annotations serving as training data. The trained Flair model is then evaluated on a validation set to assess its performance in predicting NER tags accurately. This step helps determine the effectiveness of the model and identify areas for improvement if necessary.

| Flair Annotations for WHO | Flair Annotations for WHAT |
| --- | --- |
|  |  |

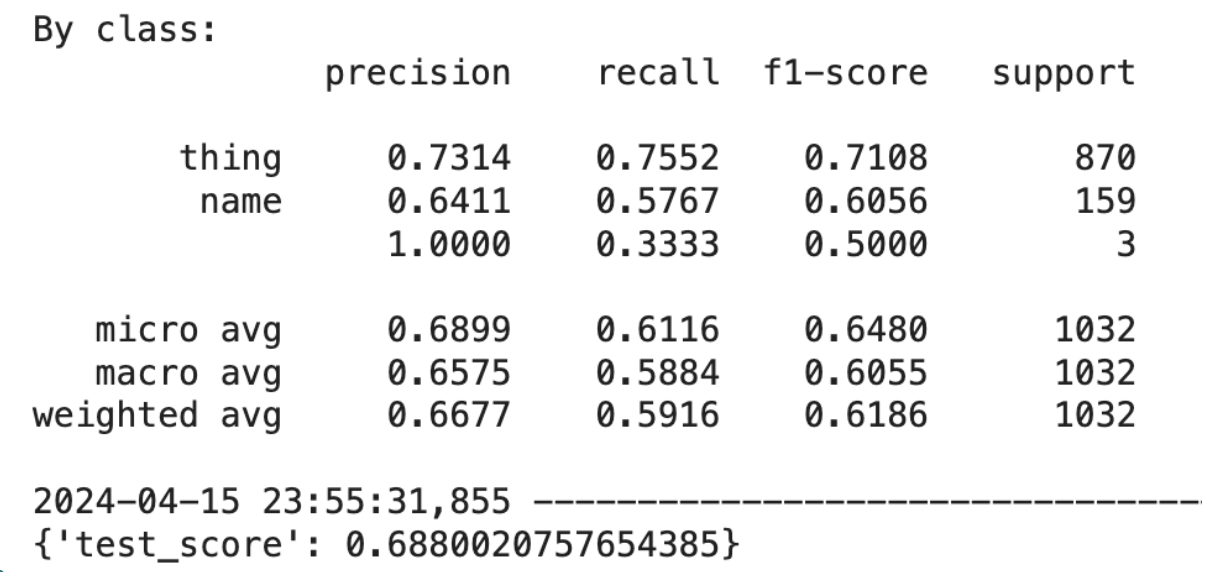
**RESULTS**

**Spacy Pipeline:**

****

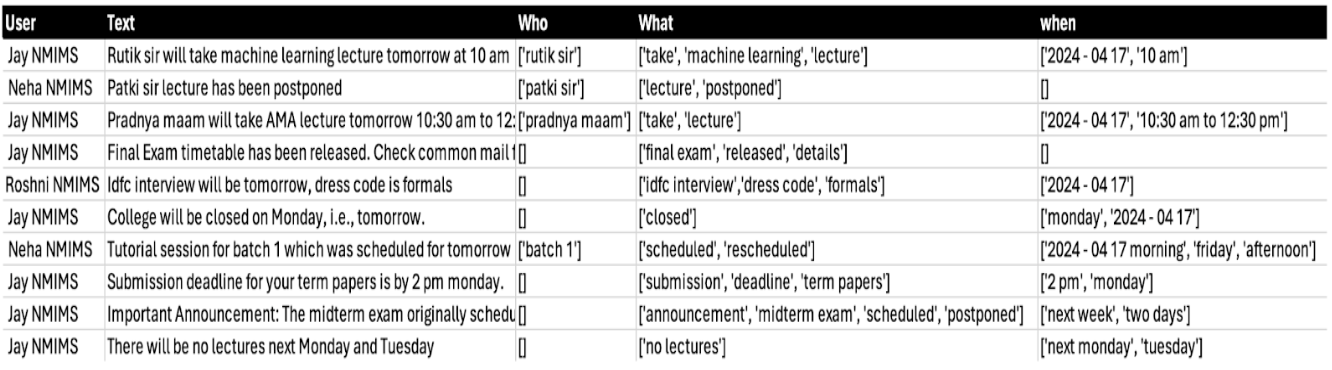
The total accuracy as well as the F-score of the Spacy Pipeline model is 95% at 600 epochs. This level of accuracy suggests that the model performs well in recognizing and categorising entities within the text data. However, it's important to note that while accuracy is a valuable metric, it may not provide a complete picture of the model's performance while considering text data, since it lacks a crucial aspect i.e., context. Our integration of the custom entities within the SpaCy Pipeline performs extremely well in terms of recognizing the entities in the conversational scope of the data- recognizing the “whats” and the “whos” however, it is important to note that without the context in picture, this remains subjective in nature. Hence, in order to also gather the context of the data we move to the Bi-LSTM+CRF model: Flair.

**Flair:**

****

Despite having a lower accuracy (of approximately 68.8%) compared to the Spacy Pipeline model, the Flair model is preferred due to its IOB (Inside, Outside, Beginning) tagging nature. The IOB nature of the Flair model provides more nuanced entity annotations.

**FINAL OUTPUT**

****

The extracted entities representing "WHO," "WHAT," and "WHEN" are consolidated into a single dataframe, along with the original sentences from which they were extracted. By appending the extracted entities along with the original sentences into single dataframe, this process streamlines the data organization and preparation for subsequent analysis or tasks, ultimately facilitating more efficient information retrieval and interpretation.

**CONCLUSION**

In conclusion, our project has successfully achieved its objectives by employing a comprehensive approach that combines advanced natural language processing (NLP) techniques, data processing methodologies, and state-of-the-art machine learning models.

Firstly, through the utilization of NLP techniques such as Spacy and RegEx, we developed a robust system capable of extracting personal information, including names, locations, organizations, dates, and events, from chat conversations. This system, coupled with a database infrastructure, facilitated efficient cataloging of crucial events mentioned within the chats.

Moreover, the integration of Flair models enriched our data processing pipeline by adding contextual embeddings and entity recognition capabilities. This enabled the extraction of nuanced information such as persons, subjects, and events, further enhancing the comprehensiveness and granularity of our extracted data.

The successful synergy between different NLP tools and machine learning models significantly bolstered the effectiveness and precision of our data extraction process. This comprehensive approach not only enhances collaboration and information management within educational institutions and organizations but also lays a solid foundation for future advancements in natural language understanding and data-driven decision-making processes.

Moving forward, continued refinement and optimization of these systems will further enhance their utility and effectiveness in addressing evolving communication and information management needs, ultimately driving innovation and efficiency in various domains.

**LIMITATIONS**

While our project has made significant strides in extracting and cataloguing information from chat conversations, several limitations should be acknowledged:

1. Limited Scope of Information Extraction: Our system primarily focuses on extracting specific types of information such as personal details, dates, and events. However, it may overlook more nuanced or contextual information present in the conversations, leading to potential gaps in data extraction.
2. Dependency on Quality of Input Data: The accuracy and effectiveness of our system heavily rely on the quality and consistency of the input data. Variations in language use, spelling errors, and informal communication styles prevalent in chat conversations can pose challenges to accurate information extraction.
3. Entity Recognition Challenges: While Flair models enhance entity recognition capabilities, they may still struggle with accurately identifying entities in complex or ambiguous contexts. This could result in misclassification or omission of relevant information.
4. Generalization to Other Domains: The performance of our system may vary when applied to chat conversations in domains different from the ones used for training and testing. Generalizing the model's effectiveness across diverse domains may require additional training and fine-tuning.
5. Scalability and Maintenance: As the volume of chat conversations grows, scalability issues may arise in managing and processing large datasets. Additionally, maintaining the system's accuracy and relevance over time requires continuous monitoring, updates, and maintenance efforts.
6. Bias and Fairness: NLP models, including Flair, may exhibit biases inherited from the training data, leading to potential disparities or inaccuracies in information extraction. Addressing bias and ensuring fairness in model predictions is crucial for the ethical deployment of the system.

**FUTURE SCOPE**

The successful implementation of our project lays the groundwork for several avenues of future exploration and enhancement. Further research and development can focus on enhancing entity recognition capabilities to accurately identify and extract a broader range of entities, including specific domain-related terms, colloquial expressions, and slang commonly used in chat conversations.

Implementing mechanisms for dynamic updation of the database and real-time extraction of information from ongoing chat conversations can enhance the timeliness and relevance of extracted insights. By continuously monitoring and processing incoming chat data, the system can provide users with up-to-date information and actionable insights as conversations unfold in real time. This capability not only ensures that users have access to the most current information but also enables proactive decision-making and response to emerging trends or events. Moreover, integrating real-time chat extraction with notification systems can alert users to important updates or discussions, further enhancing communication and collaboration within the chat environment. This feature would significantly elevate the utility and effectiveness of the system, making it indispensable for managing and extracting insights from dynamic and rapidly evolving chat conversations.

Moreover, integrating multimodal analysis techniques to analyze not only text but also accompanying images, videos, and audio clips shared within chat conversations can provide richer insights and enhance the comprehensiveness of information extraction.

**BIBLIOGRAPHY**

1. <https://dl.acm.org/doi/10.1145/3159652.3160588>
2. <https://arxiv.org/abs/1508.01991>
3. <https://arxiv.org/pdf/2109.08079v4.pdf>
4. <https://arxiv.org/abs/1603.01354>
5. <https://arxiv.org/abs/1702.02098>
6. <https://aclanthology.org/N19-4010/>
7. <https://arxiv.org/abs/1911.12237>

**CODES**

<https://drive.google.com/drive/folders/1sLHY1oJVOWalBrXsNPfn65GHWiNBKIct?usp=sharing>