Available online at: http://www.ijmtst.com/vol5issue11.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 05, Issue No: 11, November 2019

Named entity recognition using AI-NLP

Pakkurthi Sai Anudeep | Dr. K Sasikala

Department of Computer Science and Engineering, Vinayaka Mission's Research Foundation, Salem, Tamil Nadu, India.

To Cite this Article

Pakkurthi Sai Anudeep and Dr. K Sasikala, "Named entity recognition using AI-NLP", *International Journal for Modern Trends in Science and Technology*, Vol. 05, Issue 11, November 2019, pp.-200-210.

Article Info

Received on 01-November-2019, Revised on 19-November-2019, Accepted on 21-November-2019, Published on 26-November-2019

ABSTRACT

Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP), which involves identifying and categorizing named entities in unstructured text data. In recent years, deep learning-based approaches such as Long Short-Term Memory (LSTM) and Conditional Random Fields (CRF) have shown impressive performance in NER. Furthermore, word embeddings have also become a popular method for representing words in NLP tasks. In this paper, we propose a comprehensive review of recent advances in NER using AI-NLP techniques, specifically LSTM, CRF, and word embeddings. We discuss the underlying principles of these techniques and their advantages in NER. We also present a comparative analysis of these approaches on benchmark datasets and highlight their strengths and weaknesses. Furthermore, we discuss some of the key challenges in NER, such as handling rare and unknown entities, and explore potential solutions using these techniques. Overall, this paper provides a comprehensive overview of NER using AI-NLP with LSTM, CRF, and word embeddings.

I. INTRODUCTION

Named Entity Recognition (NER) is to extract valuable information from unstructured text data. In today's digital world, enormous amounts of data are generated in the form of text, such as news articles, social media posts, emails, customer feedback, and product reviews. NER helps to automate the process of extracting relevant information from these vast amounts of text data.

Named Entity Recognition (NER) is a Natural Language Processing (NLP) technique used to identify and extract named entities from unstructured text data. The main purpose of NER using AI NLP is to automatically identify and classify named entities such as people, organizations, locations, dates, time, and other important information from large volumes of unstructured text data. The output of NER is a structured representation of text data, which helps in understanding the meaning of the text and its context. NER is widely used in various applications, such as information extraction, sentiment analysis, question-answering systems, machine translation, and many more.

For example, in the financial industry, NER can be used to extract information about companies, stocks, and financial transactions from news articles and social media. In healthcare, NER can be used to extract information about diseases, symptoms, treatments, and medications from medical records and research papers. In legal documents, NER can be used to extract information about parties, dates, and legal

concepts. NER can be performed using rule-based or machine learning-based approaches. Rule-based approaches use predefined patterns and rules to identify named entities, while machine learning- based approaches train models on labeled data to automatically identify named entities.

II. NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing is a branch of artificial intelligence that attempts tobridge that gap between what a machine recognizes as input and the human language.

The goal is a computer capable of "understanding" the contents of documents, includingthe contextual nuances of the language within them. The technology can then accuratelyextract information and insights contained in the documents as well as categorize and organize the documents themselves.

Natural Language Processing consists of some techniques that are help to extract entities.NLP involves a variety of techniques and technologies, including computational linguistics, machine learning, and deep learning. Some common applications of NLPinclude speech recognition, machine translation, sentiment analysis, chatbots, and text summarization.

NLP has numerous challenges, such as dealing with the complexity and ambiguityof natural language, handling different languages and dialects, and understanding contextand sarcasm. However, as technology continues to advance, …

NLP consists of some techniques such as Tokenization, Lemmatization, Stemming, Word Embeddings, Speech Segmentation, Parts of speech tagging etc.

The top 7 techniques Natural Language Processing (NLP) uses to extractdata from text are:

1. Sentiment Analysis
2. Named Entity Recognition
3. Summarization
4. Topic Modeling
5. Text Classification
6. Keyword Extraction
7. Lemmatization and stemming

1. Sentiment Analysis:

This is the dissection of data (text, voice, etc.) in order to determine whether it's positive, neutral, or negative. sentiment analysis can transform large archives of customer feedback, reviews, or socialmedia reactions into actionable, quantified results. These results can then be analyzed for customer insight and further strategic results.

2. Named Entity Recognition:

Named Entity Recognition, or NER (because we in the tech world are huge fans of ouracronyms) is a Natural Language Processing technique that tags 'named identities' within text and extracts them for further analysis. As you can see in the example below, NER is similar to sentiment analysis. NER, however, simply tags the identities, whether they are organization names, people, propernouns, locations, etc., and keeps a running tally of how many times they occur within adataset.

Figure-1.1:Named Entity Recognition

3. Text Summary:

Text summarization is the breakdown of jargon, whether scientific, medical, technical or other, into its most basic terms using natural language processing in order to make it more understandable.

This might seem daunting – our languages are complicated. But by applying basic noun-verblinking algorithms, text summary software can quickly synthesize complicated language to generate a concise output.

4. Topic Modeling:

Topic Modeling is an unsupervised Natural Language Processing technique that utilizesartificial intelligence programs to tag and group text clusters that share common topics.

You can think of this a similar exercise to keyword tagging, the extraction and tabulation ofimportant words from text, except applied to topic keywords and the clusters of informationassociated with them.

5. Text Classification:

Text classification is the organizing of large amounts of unstructured text (meaning the raw text data you are receiving from your customers). Topic modeling, sentiment analysis, and keyword extraction (which we'll go through next) are subsets of text classification.

Text classification takes your text dataset then structures it for further analysis. It is often used to

mine helpful data from customer reviews as well as customer service slogs.

6. Keyword Extraction:
Keyword extraction is the automated process of extracting the most relevant information from text using AI and machine learning algorithms.
You can mold your software to search for the keywords relevant to your needs – try it outwith our sample keyword extractor.
7. Lemmatization and Stemming:
Lemmatization and stemming refers to the breakdown,tagging, and restructuring of text data based on either root stem or definition.
That might seem like saying the same thing twice, but both sorting processes can lend different valuable data. Discover how to make the best of both techniques in our guide to TextCleaning for NLP.

III. BACKGROUND

The theoretical background of NLP (Natural Language Processing) , LSTM(Long Short Term Memory) and CRF(Conditional Random Fields) .
Natural Language Processing (NLP) has been developed over many decades by a large community of researchers and scientists.At a high level, the theoretical foundations of NLP can be grouped into two main areas:computational linguistics and machine learning.
Computational linguistics involves the application of computational techniques to study the structure, meaning, and use of natural language. It encompasses areas such as syntax (the study of sentence structure), semantics (the study of meaning), pragmatics (the study of language use in context), and discourse analysis (the study of larger units of language, such asconversations and narratives).
Machine learning, on the other hand, involves the use of statistical and probabilistic models to automatically learn patterns in language data, such as the relationships between words or the meanings of sentences.
LSTM was introduced by Hochreiter and Schmidhuber in 1997 as a solution to the vanishing gradient problem in RNNs.In the context of NER, LSTM can be used to learn a mapping between a sequence of words in a sentence and the corresponding named entity labels. The input to the LSTM is a sequence of word embeddings, which are low-dimensional representations of the words in the sentence. The LSTM processes the sequence of embeddings one by one, updating its hidden state at each time step, and produces an output at the end of the sequence. The output is then passed through a fully connected layer that produces a probability distribution over the possible named entity labels.
CRF (Conditional Random Fields) was first introduced by John Lafferty, Andrew McCallum, and Fernando Pereira in their 2001 paper "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data." The authors proposed CRF as an improvement over Hidden Markov Models (HMM) and Maximum Entropy Markov Models (MEMM) for sequence labeling tasks, and demonstrated its effectiveness on several natural language processing tasks, including part-of-speech tagging and named entity recognition. Since then, CRF has become a popular model for many other sequence labeling tasks in various domains, including computer vision, bioinformatics, and speech recognition.

Existing System:
A rule-based system for named entity recognition (NER) involves using a set of pre-defined rules to identify and extract entities from text. These rules are typically based on patterns in the text, such as specific combinations of words or syntactic structures, and are designed to capture common patterns of entities in the text. For example, a rule-based system for NER might have a rule that looks for words that indicate a person'sname, such as "Mr.", "Mrs.", or "Dr.", followed by a sequence of capitalized words. Another rule mightlook for words that indicate a location, such as "in", "at", or "near", followed by a capitalized word.
One advantage of a rule-based system is that it can be designed to be very precise and tailored to the specific domain or task at hand. However, it can also be difficult to account for all possible variations and exceptions in the text, and the rules may need to be updated or refined over time as new patterns and variations are encountered.
A rule-based system for NER can be a useful approach in certain situations, but it may not be as robust or flexible as other methods, such as machine learning-based approaches, which can learn from data and adapt to new patterns in the text.

Disadvantages of Existing System:
Rule-based systems for Named Entity Recognition (NER) have some disadvantages:

1. Limited flexibility: Rule-based systems rely on a set of predefined rules that are manually created by domain experts. These rules may not be able to capture all possiblevariations and combinations of named entities in a given text, and may not adapt well to new types of entities that were not anticipated when the rules were created.
2. Time-consuming and costly: Creating and maintaining rule-based systems requires significant time and effort from domain experts. The rules need to be constantly updatedand revised as new entities and variations are identified, which can be time-consumingand costly.
3. Difficulty in handling ambiguity: Named entities can be ambiguous, and can have multiple meanings depending on the context in which they appear. Rule-based systemsmay struggle to accurately identify entities in such cases, especially if the rules do not account for all possible contexts.
4. Language-dependent: Rule-based systems are typically designed for a specific languageand may not perform as well in other languages or when dealing with multilingual text.
5. Limited scalability: Rule-based systems may not be scalable to handle large amounts oftext, since the rules need to be applied to each individual word in the text. This can leadto slow processing times and may not be practical for large-scale applications.
6. Difficulty in handling noise: Text data often contains noise, such as misspellings, typos,and other errors. Rule-based systems may struggle to accurately identify named entitiesin such noisy text data, especially if the rules are not designed to account for these typesof errors.
7. Limited generalizability: Rule-based systems are highly dependent on the domain expertise of the developers who create the rules. This means that the system may not be easily adaptable to new domains or applications without significant rework of the rules.

Proposed System:

To overcome the drawbacks of Rule Based System we introduced NLP techniques and LSTM (Long Short-Term Memory) for better performance. LSTM and NLP techniques can produce high accuracy for named entity recognition tasks, especially when compared to traditional rule-based or statistical methods.

LSTM-based models have the ability to handle sequential data and capture long-term dependencies between the words in a sentence, which is crucial for named entity recognition. LSTM-based models can also handle variations in the length of the input sequence, making them suitable for processing unstructured text data. LSTM is well-suited for this task because it can effectively model the sequential nature of text data, which is critical in identifying named entities.

NLP techniques like word embeddings, part-of-speech tagging, and named entity recognition can also contribute to the accuracy of named entity recognition tasks. Word embeddings can capture the semantic similarity between words, while part-of-speech tagging can provide useful information about the grammatical structure of a sentence. Named entity recognition can be used to extract relevant entities such as names, locations, and organizations from the input text.

Advantages of Proposed System:

LSTM and NLP techniques have several advantages in named entity recognition (NER) tasks:

1. Ability to handle sequential data: LSTM models can handle the sequential nature of text data and can capture long-term dependencies between words in a sentence, which is critical for NER tasks.
2. Improved accuracy: NLP techniques such as word embeddings and named entity recognition can significantly improve the accuracy of NER systems. Word embeddings can capture the semantic similarity between words, while named entity recognition can identify entities such as names, locations, and organizations with high accuracy.
3. Flexibility: LSTM models can handle variable length input sequences, which makes them flexible and well-suited for processing unstructured text data.
4. Adaptability: LSTM models can adapt to different languages and domains by training on labeled data specific to the language and domain of interest.
5. Generalization: LSTM models can generalize well to unseen data, meaning they can make accurate predictions on new text data that was not present in the training set.
6. Efficiency: LSTM models can process large amounts of text data efficiently, making them suitable for processing large datasets.

IV. System Analysis

4.1 Specifications:

Natural Language Processing (NLP) is a subfield of Artificial Intelligence (AI) that focuseson enabling machines to understand, interpret, and generate human language. The specifications on NLP include:

4.1.1 Functional Specifications:

1. Language Modelling: NLP requires developing language models that can accuratelypredict the next word in a sentence or generate new text based on a given context.
2. Machine Translation: NLP is used for machine translation, which involves translatingtext from one language to another, preserving the meaning and context of the originaltext.
3. Dialogue Systems: NLP is used for developing dialogue systems or chatbots, whichcan simulate human-like conversations with users.
4. Information Retrieval: NLP is used for information retrieval, which involves searchingand retrieving relevant information from large collections of unstructured data such astext documents, emails, and social media posts.
5. Question Answering: NLP is used for question answering, which involves answeringquestions posed by users in natural language.
6. Sentiment Analysis: NLP is used for sentiment analysis, which involves identifyingthe emotional tone of a piece of text, such as positive, negative, or neutral.
7. Text Processing: NLP systems must be able to handle and process text data in variousformats, including unstructured text data such as emails, web pages, social media posts, and other types of digital content.
8. Information Retrieval: NLP systems must be able to search and retrieve relevant information from large volumes of unstructured text data, including text documents,emails, and social media posts.

4.1.2 Non-Functional Specifications:

1. Performance: NLP systems should be designed to process large volumes of data quickly and efficiently, with minimal delays or response times, and should be scalable to handleincreased volumes of data.
2. Accuracy: NLP systems should be accurate in identifying entities, relationships,
3. concepts, and sentiments expressed in text data, with a high degree of precision and recall.
4. Security: NLP systems should be designed with robust security features, including encryption,access control, and data privacy, to protect sensitive data and prevent unauthorized access or breaches.
5. Accessibility: NLP systems should be accessible to users with disabilities, includingfeatures such as text-to-speech, voice recognition, and other assistive technologies
6. Cost: NLP systems should be cost-effective, with reasonable development and maintenance costs, and should provide value for money to users and stakeholders.
7. Compliance: NLP systems should comply with relevant legal and ethical standards, including data protection, privacy, and fairness, and should be transparent in their operations and decision-making processes.
8. User Experience: NLP systems should be user-friendly, with clear and concise interfaces that are easy to navigate, and should provide accurate and meaningful insights tousers.
9. Adaptability: NLP systems should be adaptable, with the ability to learn from new data and adjust to changing contexts or user needs, and should be able to integrate with othersystems or platforms as required.
10. Multilingualism: NLP systems should be designed to handle multiple languages, withthe ability to accurately identify and analyze text data in different languages.

4.4 Software Requirements:

- Technologies : Python
- IDE : PyCharm
- Operating System : Windows 10
- Browser: Microsoft Edge
- Libraries : spaCy ,regex, flask ,pandas , en-core-web-sm

4.5 Hardware Requirements:

- Processor : Intel Core i5
- RAM : 4.00 GB and above
- Hard Disk: 512 GB
- I/O : Keyboard, Mouse, Monitor

4.6 Module Description:

Named Entity Recognition (NER) is a Natural Language Processing (NLP) task that involves identifying and extracting entities from text, such as names of people, organizations, locations,dates, and other types of information. There are various modules that can be used for NER in AI NLP, including:

4.6.1 Preprocessing:

Preprocessing is a crucial step in Natural Language Processing (NLP) that involves cleaning and transforming raw text data into a format that can be used for machinelearning algorithms. The goal of preprocessing is to extract useful features from the raw text data while removing unnecessary noise that may hinder the performance of the algorithms.

Preprocessing can reduce dimensionality: Text data is often high-dimensional, with a large number of features. Preprocessing techniques like stemming and stop word removal can reducethe number of unique features, making it easier for machine learning models to process the dataand reducing the risk of overfitting.

Preprocessing can improve accuracy: By removing noise and reducing dimensionality, preprocessing can improve the accuracy of machine learning models by allowing them to focus on the most important features in the data.

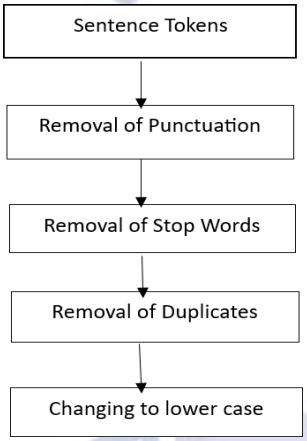

Figure-4.1: Pre-processing Flow Diagram There are several important preprocessing steps in NER, including.

Tokenization: The first step in preprocessing is to split the input text into individual words or tokens. This is typically done using a tokenizer, which is a specialized tool that can handle common challenges such as punctuation, contractions, and special characters.

Text normalization: The next step is to normalize the input text by converting it to a consistent format. This can include tasks such as lowercasing all text, removing stop words, expanding abbreviations, and correcting misspellings.

Part-of-speech tagging: Another important preprocessing step is to assign a part-of-speech (POS) tag to each word in the input text. POS tags provide information about the grammatical role of each word in a sentence, which can be useful for identifying named entities.

Entity normalization: Entity normalization is the process of mapping different surface forms of an entity to a single canonical representation. For example, the surface forms "New York City", "NYC", and "the Big Apple" might all be mapped to the same canonical representation "New York City".

Feature engineering: Finally, preprocessing may involve feature engineering, which is the process of extracting additional features from the input text that can be used to improve the performance of the classification model. Examples of features that may be extracted include word embeddings.

4.6.2 Word Embeddings:

Word embeddings are a powerful technique used in natural language processing (NLP) for representing words as vectors in a high-dimensional space. These vectors are designed to capture the semantic and syntactic meaning of words, which makes them useful for many NLP tasks, including named entity recognition (NER).

In NER, word embeddings can be used as features to represent words in the input text. This allows the NER model to capture the contextual meaning of words and the relationships between them, which is important for identifying named entities.

To use word embeddings in NER, the text is first tokenized into individual words, and each word is then represented by its corresponding embedding vector. These embedding vectors can be generated using various methods, such as Word2Vec, GloVe, or FastText.

The embedding vectors are then fed into an NER model, such as a CRF (Conditional Random Field) or an LSTM (Long Short-Term Memory) model, which uses the embeddings along with other features to predict named entity labels.

One advantage of using word embeddings in NER is that they can capture subtle relationships between words that are important for identifying named entities. For example, the embeddings of the words "New York" and "Manhattan" are likely to be similar, since they are both locations in the same city. By using these embeddings as input to the classification model, the model can learn to recognize these relationships and make more accurate predictions.

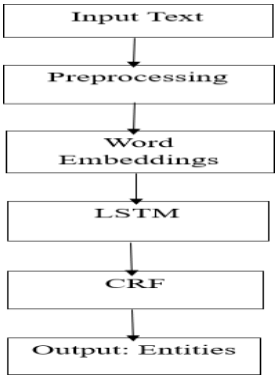

Figure-4.2 Process of NER

The input to the system is a raw text . This is first preprocessed to transform the input text into a format that can be used by the LSTM network. The preprocessing step may include tokenization, text normalization, part-of-speech tagging, entity normalization, and feature engineering, among others.

The LSTM network is a type of recurrent neural network (RNN) that is capable of capturing long-term dependencies in sequential data, such as text. The LSTM network is trained on annotated data to predict the probability of each token being a named entity. The output of the LSTM network is a sequence of probabilities for each token in the input text.

The output of the LSTM network is then fed into a CRF, which takes into account the dependencies between adjacent tokens. The CRF models the joint probability distribution over the entire sequence of output tags, taking into account the constraints imposed by the neighboring tags. This helps to ensure that the output tags are consistent with the context of the surrounding tokens.

Finally, the output tags from the CRF are used to identify the named entities in the input text. The output tags may include labels such as "person", "organization", "location", or other relevant categories. The identified named entities can then be used for various downstream applications, such as information retrieval or natural language understanding.

V. DESIGN

Software design is the process of envisioning and defining software solutions to one or more sets of problems. One of the main components of software design is the software requirements analysis (SRA). SRA is a part of the software development process that lists specifications used in software engineering. If the software is "semi-automated" or user centred, software designmay involve user experience design yielding a storyboard to help determine those specifications. If the software is completely automated (meaning no user or user interface), a software design maybe as simple as a flow chart or text describing aplanned sequence of events. There are also semi-standard methods like Unified Modelling Language and Fundamental modelling concepts. In either case, some documentation of the plan is usually the product of the design. Furthermore, a software design may be platform-independent or platform-specific, depending upon the availability of the technology used for the design.

Software design is both a process and a model. The design process is a sequence of steps that enables the designer to describe all aspects of the software for building. Creative skill, past experience, a sense of what makes "good" software,and an overall commitment to quality are examples of critical success factors fora competent design. It is important to note, however, that the design process is not always a straightforward procedure; the design model can be compared to an architect's plans for a house. It begins by representing the totality of the thing thatis to be built (e.g., a three-dimensional rendering of the house); slowly, the thing is refined to provide guidance for constructing each detail (e.g., the plumbing lay).Similarly, the design model that is created for software provides a variety of different views of the computer software. Basic design principles enable the software engineer to navigate the design process.Rules could be created to capture common prefixes and suffixes used in person's names, such as "Mr.," "Ms.," "Jr.," "Sr.," and so on. These rules can be used to extract the person's names from the identified entities.

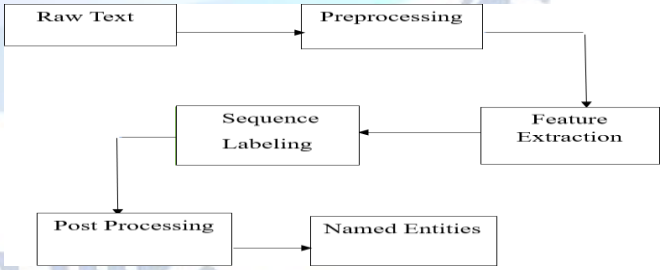

Figure - : Block Diagram for Named Entity Recognition

VI. IMPLEMENTATION

Our experimental setup consists of Python Script for processing text, source text iscollected from various sources like news articles, blogs, articles and so on. It is then, pre-processed by using sentence tokenization, removing irrelevant data

suchas frequently words in English (referred as stop words) and punctuations, which induce noise in the construction process.
Individual sentences are compared to deduce similarity with the help of measuressuch as word overlap, TF-IDF (referred as Term Frequency – Inverse Document Frequency) statistics. Similarity scores are normalized for better accuracy. We then,construct a graph out of the resulting vectors and apply the PageRank Algorithm, obtaining a set of scores for each sentence. Taking into account of length of the sentences, we consider the highest ranked sentences, representing significant information, for construction the final output. article_id, article_text and sources are the fields that are considered for our processing the output. Sources is the fieldwhere we have various source link from where that data is taken for processing.

VII. RESULTS

7.1 Person:
Rules could be created to capture common prefixes and suffixes used in person's names, such as "Mr.," "Ms.," "Jr.," "Sr.," and so on. These rules can be used to extract the person's names from the identified entities.
Here in the below figure in the text box give some textual data then select the task as person then the NER app will extract the person's names in the given input text.

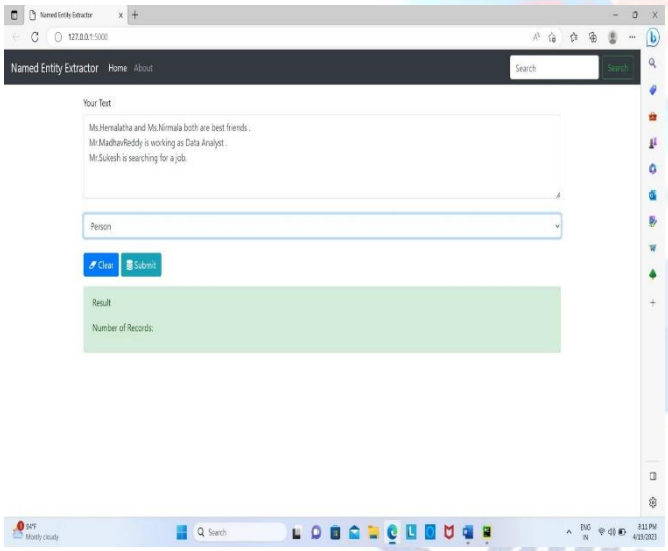

Figure 7.1: Input Text to Extract Person's names

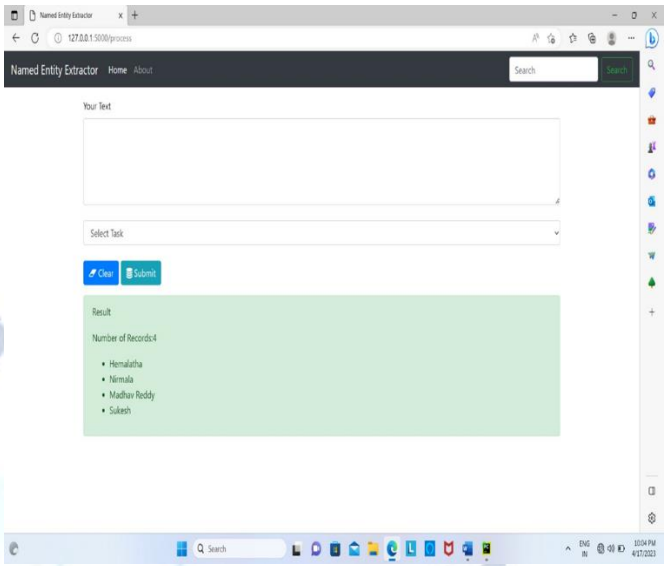

Figure-7.2: Extracted Person's Names The above figure shows the extracted person's names from the user input.

7.2. Organization:
Our NER app will extract the organization names from the given input data. Here we given some data which contains some organizations. The model learns to recognize patterns and characteristics of organization names in text and can then be used to extract the organization names from the identified entities.
After given the input text select task that and click on submit button .

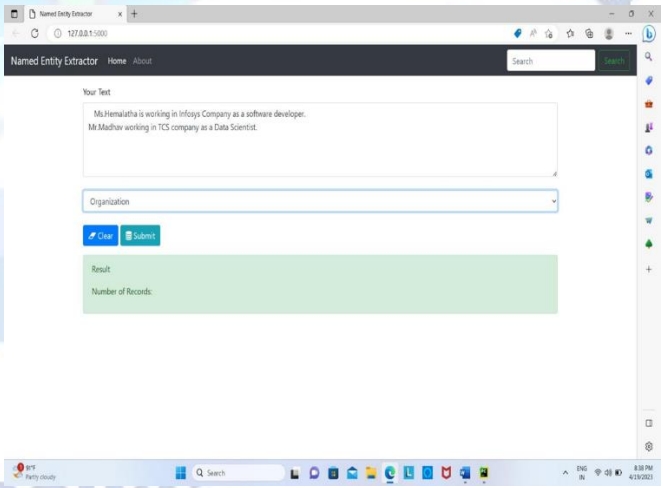

Figure-7.3:Input Text to Extract Organization Names

Here in the below figure, it extracted the organization's names.

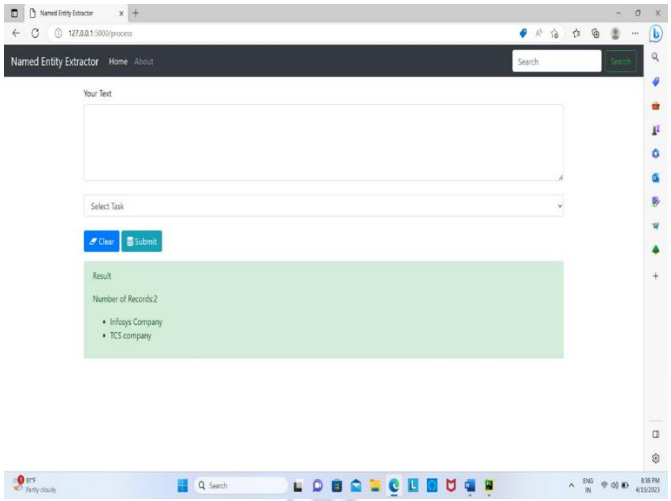

Figure-7.4:Extracted Organizations Names

7.3. Geopolitical:

Geopolitical entities can include countries, cities, states, regions, and other geographical entities. Once the named entities have been identified, various techniques can be used to extract the geopolitical names from the identified entities.

rules could be created to capture words that commonly appear in geopolitical names, such as "United," "States," "Republic," "Province," and so on. These rules can be used to extract the geopolitical names from the identified entities.

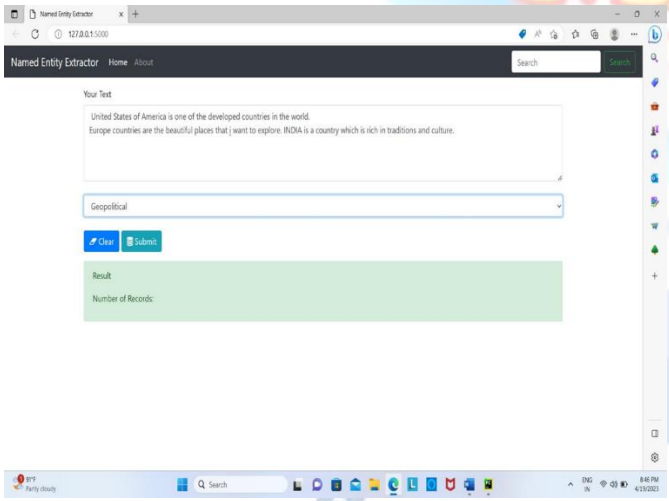

Figure-7.5:Input Text to Extract Geopolitical Entities

In the above figure the text is given and geopolitical task is selected. Then NER app will extract the locations or places names.

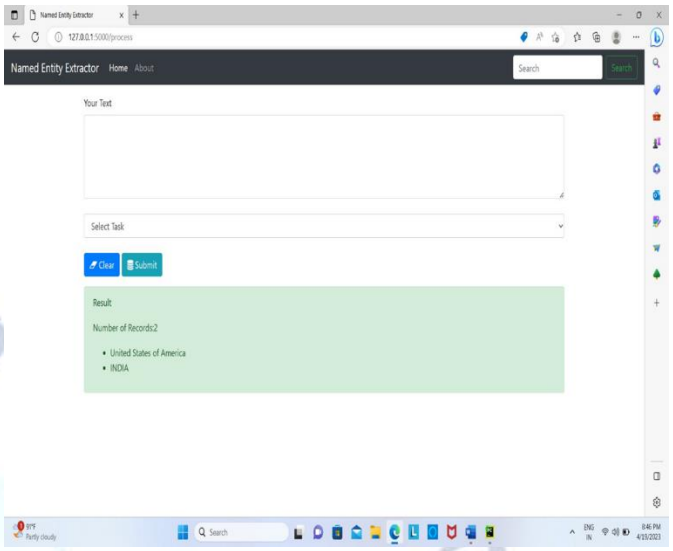

Figure-7.6:Extracted Geopolitical Names

7.4 Money:

Money entities can include currency symbols, numeric values, and other symbols associated with money, such as "$," "USD," "dollars," and so on. Rules could be created to capture words that commonly appear with money amounts, such as "million," "billion," "thousand," and so on. These rules can be used to extract the money amounts from the identified entities.

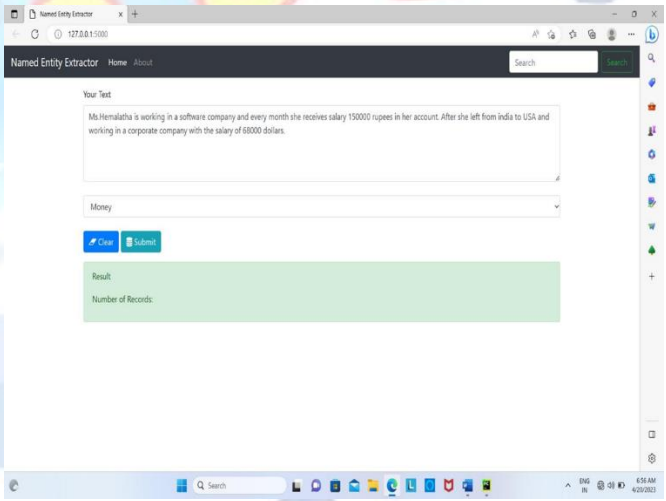

Figure-7.7:Input text to Extract Money

Here in the below figure it extracted the money from the input which is given by the user.

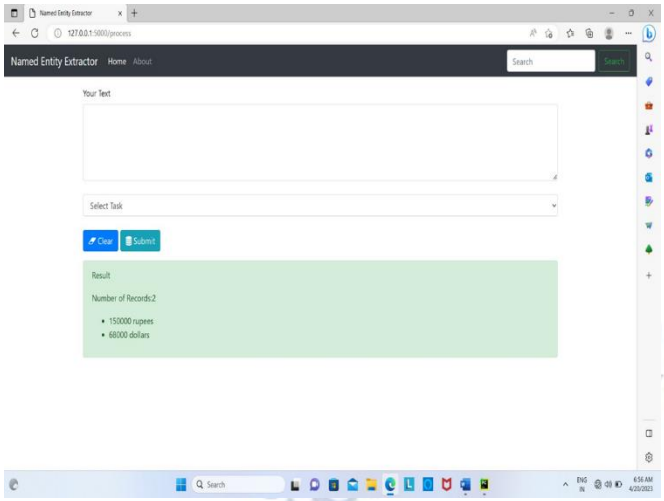

Figure-7.8:Extracted Money

VIII. CONCLUSIONS AND FUTURE SCOPE

Named Entity Recognition (NER) is an important task in Natural Language Processing (NLP)that involves identifying and extracting named entities such as people, organizations, locations,dates, and other entities from text data.In our project we done NER app to extract the entities like person's names, organization's names, Geopolitical areas names and it can also extract the Money which is declared with its units.

We take text data whether it is structured or unstructured then we have to select any option like person, organization, geopolitical, money based on our requirement. Here in NER, we used NLP techniques,LSTM,CRF.

NER using AI-NLP has various applications, including information extraction, sentiment analysis, recommendation systems, and chatbots. It can help in automating processes, improving customer experiences, and reducing errors in various industries, including healthcare, finance, legal, and e- commerce.

In conclusion, NER using AI-NLP is an exciting field with tremendous potential for future applications. It has already revolutionized many industries and will continue to do so in the future. As AI-NLP technology advances, we can expect even more accurate and efficient NER models to be developed, enabling more sophisticated applications of this technology.

FUTURE SCOPE

Here are some potential future enhancements of NER that we want to execute in future:

In the future scope of this project, we will try to implement the NER, inputs in the form of text images, images and by uploading files.

Multilingual Entity Recognition: NER systems are typically designed to recognize named entities in a specific language. However, as more and more data become available in multiple languages, multilingual entity recognition will become increasingly important. This involves developing NER models that can recognize named entities in multiple languages.

Domain-Specific Entity Recognition: NER systems can be tailored to specific domains, such as legal or medical texts, where the entities of interest may be different from general texts. Developing domain- specific NER systems can improve accuracy and performance.

Relation Extraction: In addition to recognizing named entities, NER can be used to identify relationships between entities. Relation extraction involves identifying the type and direction of the relationship between entities (e.g., person A is the CEO of organization B). Developing NER systems that can perform relation extraction can enable more advanced NLP applications such as knowledge graph construction and event extraction.

REFERENCES

[1] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

[2] Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32, 5754-5764.

[3] Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. Proceedings of the 27th International Conference on Computational Linguistics, 1638-1649

[4] Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2018). Semi-supervised sequence tagging with bidirectional language models. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 1756-1765

[5] Ma, X., & Hovy, E. (2018). Jointly tagging entities and relations with a transformer-based model. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 3678- 3687

[6] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[7] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. Proceedings of the 2016 Conference of the North American Chapter of the Association for

Computational Linguistics: Human Language Technologies, 260-270.

[8] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

[9] Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354.