**Software Requirements Specification (SRS)**

**Project Name :** Named Entity Recognition

Prepared by Abhishek Soni

Date : 26 / 02 / 2025

**Table of content :**

1. Overview
2. Subject
3. Objective
4. How Named Entity Recognition(NER) works?
5. Why is this Named Entity Recognition needed?
6. How are these entities predicted?
7. Continuous Learning
8. Integrate with Streamlit

**1. Overview :**

**Industry Type :** Natural Language Processing Industry

The **Natural Language Processing (NLP)** industry is a part of the artificial intelligence (AI) industry that focuses on enabling computers to understand and interact with human language.

NLP is used in many industries,including healthcare,finance, retail, education, and many more.

**2. Subject** : Named Entity Recognition

**Name-entity recognition (NER)** is also referred to as **entity identification**, **entity chunking,** and **entity extraction**. NER is the component of information extraction that aims to identify and categorize named entities within unstructured text. NER involves the identification of key information in the text and classification into a set of predefined categories.

An entity is the thing that is constantly talked about or referred to in the text, such as person names, organizations, locations, time expressions, quantities, percentages and more predefined categories.

**3. Objective :**

**Named Entity Recognition (NER)** is a technique in **natural language processing (NLP)** that focuses on identifying and classifying entities. The purpose of NER is to automatically extract structured information from unstructured text, enabling machines to understand and categorize entities in a meaningful manner for various applications like text summarization, building knowledge graphs, question answering, and knowledge graph construction.

* Detecting the entities from the text
* Classifying them into different categories

**4. How Named Entity Recognition (NER) works?**

The working of Named Entity Recognition is discussed below:

* The NER system analyses the entire input text to identify and locate the named entities.
* The system then identifies the sentence boundaries by considering capitalization rules. It recognizes the end of the sentence when a word starts with a capital letter, assuming it could be the beginning of a new sentence. Knowing sentence boundaries aids in contextualizing entities within the text, allowing the model to understand relationships and meanings.
* NER can be trained to classify entire documents into different types, such as invoices, receipts, or passports. Document classification enhances the versatility of NER, allowing it to adapt its entity recognition based on the specific characteristics and context of different document types.
* NER employs machine learning algorithms, including supervised learning, to analyze labeled datasets. These datasets contain examples of annotated entities, guiding the model in recognizing similar entities in new, unseen data.
* Through multiple training iterations, the model refines its understanding of contextual features, syntactic structures, and entity patterns, continuously improving its accuracy over time.
* The model’s ability to adapt to new data allows it to handle variations in language, context, and entity types, making it more robust and effective.

**5.Why is this Named Entity Recognition needed?**

Named Entity Recognition (NER) is needed because it allows machines to automatically identify and categorize key entities like people, places, and organizations within unstructured text, effectively transforming raw text into structured data that can be easily analyzed and used for various applications, like information extraction, question answering, and text summarization, providing valuable insights from large amounts of text without manual effort.

**6.How are these entities predicted?**

### **Rule Based Method**

The Rule Based NER method uses a set of predefined rules guides the extraction of information. These rules are based on patterns and context. Pattern-based rules focus on the structure and form of words, looking at their patterns. On the other hand, context-based rules consider the surrounding words or the context in which a word appears within the text document. This combination of pattern-based and context-based rules enhances the precision of information extraction in Named Entity Recognition (NER).

### **Machine Learning-Based Method**

#### **Multi-Class Classification with Machine Learning Algorithms**

* One way is to train the model for [multi-class classification](https://www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/) using different machine learning algorithms, but it requires a lot of labelling. In addition to labelling the model also requires a deep understanding of context to deal with the ambiguity of the sentences. This makes it a challenging task for a simple machine learning algorithm.

### **Deep Learning Based Method**

* Deep learning NER system is much more accurate than the previous method, as it is capable of assembling words. This is due to the fact that it used a method called word embedding, that is capable of understanding the semantic and syntactic relationship between various words.
* It is also able to analyze topic specific as well as high level words automatically.
* This makes deep learning NER applicable for performing multiple tasks. [Deep learning](https://www.geeksforgeeks.org/introduction-deep-learning/) can do most of the repetitive work itself, hence researchers for example can use their time more efficiently.

For all this we can refer to dataset from kaggle

Reference 1 : <https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus/data>

Reference 2 :

<https://www.kaggle.com/datasets/naseralqaydeh/named-entity-recognition-ner-corpus/data>

**Use pre trained models :**

**‘bert-base-ner’ model of Hugging-face :**

bert-base-NER is a fine-tuned BERT model that is ready to use for Named Entity Recognition and achieves state-of-the-art performance for the NER task. It has been trained to recognize four types of entities: location (LOC), organizations (ORG), person (PER) and Miscellaneous (MISC)

**‘En-core-web-sm’ of spacy :**

The term "en-core-web-sm" refers to a pre-trained Natural Language Processing (NLP) model in spaCy, a popular open-source library for NLP tasks. Specifically, the "en-core-web-sm" model is a smaller, English language model that is optimized for performance and speed, making it ideal for use in environments with limited resources.

Here's a breakdown of the components:

* en: Stands for English, indicating that the model is designed for the English language.
* core: Indicates that it is a general-purpose model that can handle a range of NLP tasks.
* web: Suggests that this model has been trained on web text, so it is optimized for understanding language found in online content (e.g., social media, blogs, websites).
* sm: Stands for "small," which means that this is a lighter, faster version of the model compared to larger ones like "en-core-web-md" (medium) or "en-core-web-lg" (large). The small version has a smaller size and fewer parameters, making it faster to load and use but potentially less accurate than the larger models.

This model can perform several NLP tasks such as:

1. Named Entity Recognition (NER): Identifying entities like people, organizations, and locations in text.
2. Part-of-Speech Tagging (POS): Assigning grammatical labels to words, such as verbs, nouns, etc.
3. Dependency Parsing: Understanding the syntactic structure of sentences.
4. Text Classification: Categorizing text into predefined categories.

**7. Continuous learning :**

"Continuous learning" is refers to the ability of a model to continuously acquire new knowledge and adapt to changing data distributions over time, without forgetting previously learned information, essentially mimicking how humans learn throughout their lives by accumulating knowledge across various situations and tasks; it's also known as "continual learning" or "lifelong learning" in the field of AI.

**8. Integrate with Streamlit :**

Streamlit is a promising open-source Python library, which enables developers to build attractive user interfaces in no time. Streamlit is the easiest way especially for people with no front-end knowledge to put their code into a web application: No front-end (html, js, css) experience or knowledge is required.