**Project Name:** Named Entity Recognition

**Github Link:** https://github.com/projectsforstudents2022/Named-Entity-Recognition.git

**Why was this project created?**

An essential NLP task is text classification, which may be thought of as a set of texts and labels. In addition to other texts, we want to construct a classifier that can classify these inputs as well. Text comprehension and phrase extraction for model training are the fundamental components of text classification problems. For many Natural Language Processing applications, which entail comprehending, extracting information from, and categorizing the text, Text Classification and NamedEntity Recognition are two important tasks.

**What problem is it solving?**

We are driven by a desire to investigate Named Entity Recognition models that are connected to language. This project, in our opinion, is a great chance to investigate language models and perform tasks like named-entity recognition and text classification. We used AutoPhrase and a pre-trained language Named Entity Recognition model to extract high-quality phrases in order to accomplish these goals.

**Entire explanation of project**

* **PROPOSED APPROACH**

Any Text Mining problem's Data Preprocessing step is the most difficult. We need to clean the data first before we use any algorithms. Therefore, we must first remove the data devoid of any tags. In order to save the data as a csv (comma separated value) file, we first clean the data so that for each document id, we only capture the document's body and any connected topics. The document's body is transformed into a bag of words and then saved as a CSV file. We construct a consolidated vocabulary, or the union of all the vocabs, after creating the individual vocabulary files for each label. As our feature vector, this is effective.

With the aid of the train test split library, we divided the data into train and test using the saved matrices. After training the model with a polynomial kernel, we utilised an RBF kernel. Accuracy, precision, and recall values are computed. The procedures outlined above must be followed, and the model must be trained using all of the training data rather than just 70% of it, in order to generate a classifier for a fresh dataset or unlabeled data. The unlabeled data would be the test data. We forecast the classification of unlabeled data using the model trained on test data.

Algorithm for creating next word prediction model :

**Step 1:** Import Libraries & Load Dataset

**Step 2:** Tokenization

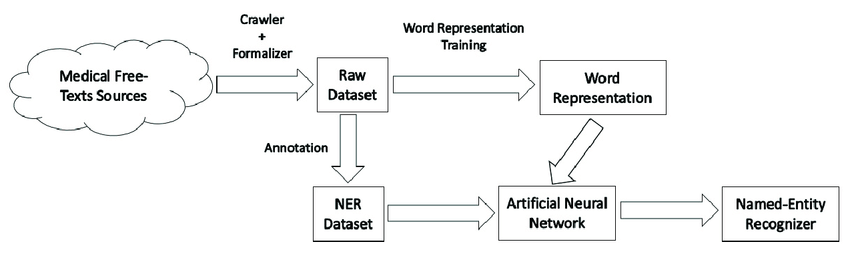
**Step 3:** Part of Speech Tagging & Chunking

**Step 4:** Build Neural Network

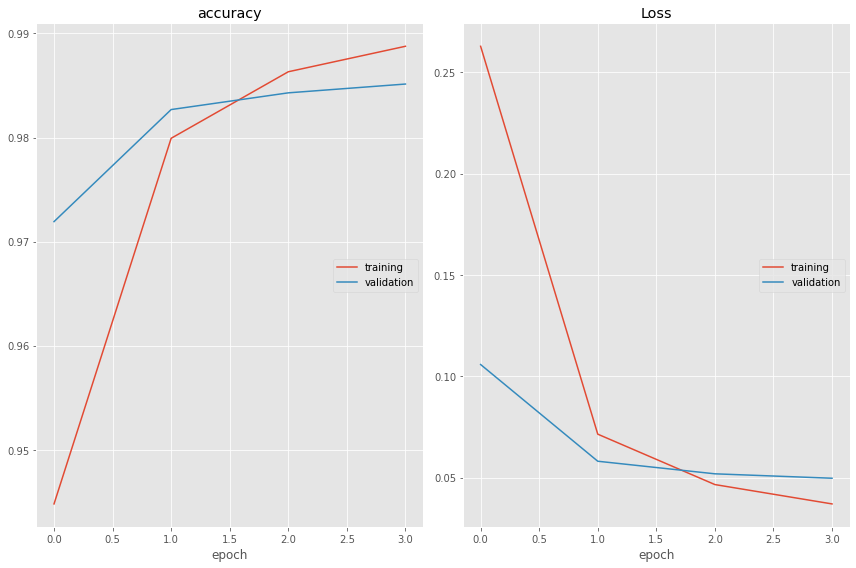
**Step 5:** Train Model

**Step 6:** Testing & Visualization

* **DATA FLOW DIAGRAM**



* **RESULT**



* **CONCLUSION**

In this research, we used a pre-trained model and AutoPhrase to classify text. The accuracy and F1 score of the results are quite strong. Entity and quality phrases are, in our opinion, particularly potent features to use in text classification tasks. However, we would obtain a far better performance if we also included the uni-gram functionality.