Named Entity Recognition for Malayalam and Hindi.

Sai Kesav.R
CB.EN.P2CEN19010
Center for excellence in
Computational Engineering
& networking
Natural Language Processing
Amrita Vishwa Vidyapeetham
Coimbatore, India
rachakondasaikesav@gmail.com

***Abstract*— Named entity recognition is one of the most reliable applications of Natural Language Processing. This report is a brief description on the two contests assigned for the term work submission of Natural Language Processing based on Named entity recognition on two two different paninian languages ,Hindi and Malayalam.**

INTRODUCTION

As more and more languages are being computationally used for various purposes, it is quite necessary to analyze them on a whole basis without being able to interpret each sentence separately. Named Entity Recognition (NER) is the process of determining the piece of writing in any specific language according to its characteristic terminology and NER helps us to analyze and identify the key elements in a text of any language based on names, places, brands, genres, monetary values and the extracting of the main entities in a text help to sort the unstructured data and detect the important information, which is actually crucial when interpreting huge datasets. Named Entity Recognition (NER) is also known as entity chunking or the extraction to identify and classify the information based on its type and the entity chosen to be identified for the chosen data and classify them under the predefined classes.

There are different methodologies to solve these kinds of NER tasks using word embeddings, tokenization, and using different mechanisms and methods like BERT, Fasttext.

For the successful completion of the assigned two contests, where the 1st task being the entity recognition of news data based on malayalam language, a simpler methodology of tokenization and a uncomplicated architecture of RNN was used for the training of the model with a k-fold cross validation of the training data for better accuracy results.

For the successful completion of the 2nd task being the entity recognition of words classified in Hindi language, word embeddings based on fasttext were utilized and a Decision Tree Classifier, Machine Learning algorithm was used to classify the words as per their defined classes for the given test data and the to predict their labels as per the classes on which they were trained given the specific training data.

LITERATURE SURVEY

An exemplary utilisation of NLTK was utilised for the completion of the contest where NLTK, the Natural Language ToolKit is a suite of open source program modules, tutorials and problem sets, which provide ready to use computational linguistic courseware[1]. NLTK enables and benefits the user with symbolic and statistical natural language processing and is interfaced to an annotation for a given corpora of words. In NLTK the classifier module defines a standard interface for classifying texts into categories and a Naive Bayes module based on the Naive Bayes classifier is used which implements Generalized Iterative scaling and Improved Iterative scaling, and this classifier feature provides a standard encoding for the information that is used to make decisions for a particular classification task, and the standard encoding allows the students to experiment with the differences between different text classification. The feature selection module defines a standard interface for choosing the relevant features for a particular classification task. Good feature selection can significantly improve the classification performance.

Li.et .al [3] generalized that Named Entity Recognition (NER) is the task to identify the text spans that mention the named entities, and to classify them into predefined categories such as person, location, genre etc. and NER not only acts as a standalone tool for information extraction, but also plays an important role in a variety of NLP applications, such as text

understanding, information retrieval, automatic text summarization, question answering and the knowledge base construction. although NER is quite capable of producing decent recognition accuracy and requires less human effort in carefully designing the features when compared to the early versions of NER. Deep learning models, empowered by continuous real-valued vector representations and uses semantic composition of the words, sentences and the text chosen, through non-linear processing, which increases the yielding of the NER and better classification of the assigned task.

In Paninian languages and Urdu based languages [5] NER becomes quite complicated but the model based on the Deep recurrent neural network (DRNN) demonstrate that they improve the approach of NER and considers the language dependent features and INLTK also supports this model, where the word embeddings enable the creation of the "context windows" of words, and the vanishing gradient problem when trained using Back propagation is solved using a framework of LSTM. In almost all approaches, the best performance relies on constructing an optimal set of features, and NER with a combination of RNN uses a varied of linguistic features ranges from semantic information on words to information on the morphological and syntactic structures of words, which enables easy learning of the labelled and unlabelled data.

Word Embeddings [5] is that words occurring close to each other should acquire the required NE labels, and these embeddings the dense features play a vital role in describing the characteristics of the given token where each token can point to a vector and the value of that vector will be used as the embedding feature, which are produced in an unsupervised manner but correspond to the grouping of similar words which capture the diverse aspects of their meanings based on their phrases within the vector representation features.

In the field of machine learning, the performance of a classifier is usually measured in terms of the prediction error [2]. Cross-validation is a resampling procedure which is used to evaluate machine learning models on a limited data sample, and this procedure has a single parameter called k that refers to the number of groups that a given data should be split into, and the utilisation of k-fold cross validation is one of the standard techniques to detect overfitting and there is no guarantee that k-fold cross-validation removes overfitting. The general method is to split the dataset into training and the testing sets, and then evaluate the model performance based on the error metric to determine the accuracy of the model, where as for better results K-fold Cross Validation provides a solution to this by dividing the data into folds and ensuring that each fold is used as a testing set at some point.

Decision tree induction was used as a solution for the problem of customising NER and classification to a specific kind of domain.[4]. where the Named Entity Recognition and Classification system assigns tags to phrases that correspond to named entities such that the system uses two language resources, one based on the recognition grammar and the other on the lexicon of known names, which are labelled and classified by the entities. A simple method was observed by Paliouras.et.al, that the recognition algorithm formed recognisers using the C4.5 decision tree can be easily translated into a small number of comprehensible rules. Rules simplicity is enhanced with the use of a special facility of subsetting of the feature values after the embedding of the words for the many valued-features, at the same time utilisation of C4.5 decision trees removes the need for a pre-processing stage, which is common in all systems which use the decision trees, and in NERC due to the advancements done by [4] Paliouras.et.al the decision tree provides direct classification using the Named Entity Recognition and Classifier under Decision trees.

Specifically in Malayalam the tokenization of words is done through the splitting of sandhi and the rules led us to a research paper [6]. Malayalam is a highly agglutinative language like most of the Dravidian languages, where as in english simple splitting with spaces and punctuations will not properly tokenize an English sentence and to prevent the same problem in malayalam they use Subword tokenization where the word embeddings for each split word the subword is tokenized[8].

FastText is an open-source, free, lightweight library that allows users to learn text representations and text classifiers. and it works on standard and generic hardware[9]. When compared with the word2vec, in which there is a limitation of rare word representations which can be explicitly learnt and the representation for the words which are inside the set vocabulary are only specifically available and it is quite to difficult to improve the vector representations for the morphologically rich languages using the character level information. Fast-text uses skip-gram with negative sampling for finding the word vector representations of context words, where negative sampling helps us to modify the small percentage of the weights, rather than all of the weights for training the sample and by this the prediction of two words is not predicted but the neighbourity and relation of two words in a sentence is measured.[8] A pre-trained word-vector model for hindi language with a distrubed dimension of 300, was used for the 2nd contest.

DATASET DESCRIPTION

For Contest-1, two datasets, training and testing dataset were assigned. With a total of 5037 sentences in Malayalam and been labelled into three classes of news entities be in sports, entertainment, and business. And a test data with a total of 1261 sentences with to be predicted labels were assigned.

For Contest-2, two datasets, training and testing dataset were assigned. With words from each representing location, occupation, name, organization, number, things, other,date-num and newline, and labelled as them. And a test data 12666 words of a sentence separated with a newline similar to that of training data, and the test data represented the tags with which the appropriate labels for each word are to be represented.

METHODOLOGY

For the completion of Contest-1, initially a simpler method of tokenizing the words in the whole training dataset was implemented with a train and test split of 70-30, and a RNN architecture was used, which resulted in a much lesser accuracy, as they were few punctuations, which were getting notified as tokens, but not the words, then a little improvisation was done by removing the punctuations and splitting the data using k-fold cross validation with 4 splits, where the data is split in the ratio of n-1 training samples and 1 testing samples randomly and repeatedly until all the iterations are completed. The nb-tokenized embeddings and the labels in a categorical format were splitted using the K-fold cross validation and were trained using a Simple RNN architecture, consisting of a single hidden layer consisting of 100 neurons and 2 dense layers with 100 and 50 neurons in each dense layer respectively, the no. of words to be considered for the model as input are 1200 including the embedded dimensions and 500 embedded dimensions of embedded values generated. The loss was calculated using categorical cross entropy and a softmax activation layer was used. Total of 675,403 trainable parameters were generated, 96.16% and 84.83% are the respective training and testing accuracies achieved.

In almost all approaches, the best performance relies on the chosen optimal features , and Named Entity Recognition with a combination of RNN uses a varied of linguistic features ranges from semantic information on words to information on the morphological and syntactic structures of words, which enables easy learning of the labelled data and gives us better predicted labels when tested against the testing data. The labels were generated and submitted for evaluation.

For the completion of contest-2, a pretrained model of fasttext embeddings based on the Hindi language and following the same morphological order were chosen[8]. and the training data consisting of various words splitted under their respective labels for each sentence were used against the chosen fasttext embeddings and were embedded into the needed vectors and the labels were splitted for training and testing in the ratio of 70-30, and a Decision Tree classifier was used for the model and an accuracy of 87% was achieved.

Fast-text uses skip-gram with negative sampling for finding the word vector representations of context words, where negative sampling helps us to modify the small percentage of the weights, which enabled us to get the prediction of the two neighbouring words in the sentence and generated better embedding vectors which resulted in better accuracy, as fasttext unlike word2vec embedding takes out of vocabulary words and generates vectors for them. and from the [4] it was certain that Named Entity Recognition and Classification when used via a Decision Tree classifier gives better predicted values, when compared to other Machine Learning algorithms, for certain specific datasets, with the required morphological and semantic structures.

RESULTS

All the predicted labels for the respective contests for the assigned test datasets and are plotted in the table1.

Contest	**Evaluator**	**Score**
01	F1-macro	74.23%
02	F1-macro	65.65%

Table 1 : Evaluated result scores of each contest

The both contests were evaluated on the basis of F1-score due to the class imbalance observed in the both datasets.

CONCLUSION

The accuracy and the F1-score and the amount of correct labels being predicted can be further increased by solving the class imbalance in the dataset and by using further tuning parameters, which may avail for the better results and correct prediction of the labels.

REFERENCES

[1]Loper, Edward & Bird, Steven. (2002). NLTK: the Natural Language Toolkit. CoRR. cs.CL/0205028. 10.3115/1118108.1118117.

[2] Rodríguez, Juan & Pérez, Aritz & Lozano, J.A.. (2010). Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 32. 569 - 575.

[3] J. Li, A. Sun, J. Han and C. Li, "A Survey on Deep Learning for Named Entity Recognition," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2020.2981314.

[4] Paliouras, G., V. Karkaletsis, Georgios Petasis and C. Spyropoulos. “Learning Decision Trees for Named-Entity Recognition and Classification.” (2000).

[5]Khan, Wahab, Ali Daud, F. S. Alotaibi, Naif R. Aljohani and S. Arafat. “Deep recurrent neural networks with word embeddings for Urdu named entity recognition.” *Etri Journal* 42 (2020): 90-100.

[6]Nisha, M. & Raj, P.C.. (2016). Sandhi Splitter for Malayalam Using MBLP Approach. Procedia Technology. 24. 1522-1527 . 10.1016/j.protcy.2016.05.113.

[7]https://blog.qburst.com/2020/04/malayalam-subword-tokenizer/

[8] https://fasttext.cc/