Utilizing a Neural Network to recognize named entities in the Malayalam language.

***Abhishek Prathap, *Adithya A Nair, *Aravind P, **Vimal Babu P**

* Computer Science and Engineering, Mangalam College of Engineering, Kottayam, Kerala
**Computer Science and Engineering, Assistant Professor, Mangalam College of Engineering, Kottayam, Kerala

Abstract

The process of recognizing named things in a text document entail classifying them into specified groups like Person, Location, Organization, etc. It is a crucial stage in the processing of text written in natural language. The goal of named entity recognition systems is to extract pertinent data from the text. Different approaches are used in Malayalam for NER. In this paper, we provide a neural network-based named entity recognition system for Malayalam. For learning representations of data with various levels of abstraction, neural networks are incredibly potent tools. The suggested approach uses various properties, including embedded word and suffix representations, POS information about the word, etc. We only used a corpus of 6300 headlines for Malayalam news articles that were gathered from Malayalam news websites. The system was able to achieve the most cutting-edge performance in NER for Malayalam with fewer features.

1. Introduction

The amount of information available on the internet is constantly growing. Every second, new words and images are uploaded to the internet, contributing to the issue of information overload. Furthermore, this data is accessible in an unstructured way. We are unable to search through all of these data for pertinent information. The practice of gathering pertinent material from an enormous quantity of unstructured data is known as information extraction in the field of artificial intelligence. By using information extraction (IE), unstructured content is transformed into a structured format that computers can process with ease. One of IE's subdomains is NER. In the sixth message understanding conference (MUC-6), the word "entity" is first used. Additionally, from MUC conferences are the benchmarks for various NER systems. Expressions that relate to specific people, places, organizations, etc. are known as named entities. Due to the Malayalam text's ambiguous formatting, entity extraction is a challenging operation. English, for example, has a specified and arranged shape for its text, making entity extraction in such languages a straightforward operation. It can be difficult to identify named entities in an open-domain unstructured text. Even though various solutions to this issue have been presented, the field of study is yet unexplored. The ultimate objective of named entity recognition systems is the identification and classification of named entities to classes with semantically meaningful names.

Building computational models for the analysis and creation of natural language text is the goal of natural language processing. These models can be used to create intelligent computer systems like summarizing systems, machine translation systems, and speech recognition systems, among others. Named entities make up about 60% of all search engine inquiries. Determining named entities from an open domain text is crucial for processing queries. Systems for recognizing named entities are useful in question-answering software as well. Finding such entities is helpful for question-answering systems because the majority of inquiries with the keyword "who" always have an entity with the class "person" as their answer.

The task of named entity recognition in Malayalam is difficult for the reasons listed below. Languages like English have the ability to capitalize words to distinguish between named entities. However, Malayalam lacks the capitalization feature, which makes it more difficult to distinguish between named entities. Identifying named entities in Malayalam is also complicated by its rich morphology. Multiple stems and affixes are frequently combined to create words in Malayalam. Identification of named entities is additionally hampered by the case information associated with noun terms. In Malayalam, a large portion of the words are agglutinated. Agglutination produces complicated words that are frequently challenging to understand. A further challenge in creating NER systems for Malayalam is the absence of linguistic resources like standard datasets, POS taggers, morphological analyzers, dictionaries, etc.

In this paper, we provide a neural network-based named entity recognition system for Malayalam. Our ultimate goal is to raise entity recognition systems' efficiency. The main entity classes that we have taken into consideration are person, place, organization, and miscellaneous. All other types of entities fall under the miscellaneous category. The format of this essay is as follows. The paper's second half provides a brief survey of related literature. The suggested approach is explained in Section 3. The experiments and findings are described in Section 4. The paper's final part offers several directions for future research.

2. Related works

One of the hot topics in NLP in recent years has been named entity recognition. Various methods for automatic named entity recognition are reported. Rule-based, machine learning-based, and hybrid technologies are the three main categories. The foundation of manually crafted rules is the foundation of rule-based systems. They are particular to the languages for which they are written and can only conduct entity recognition in constrained

domains. Numerous rules are needed for rule-based systems to identify named entities, and language experts are also required to write the rules. They are not robust or portable. Therefore, researchers in this field focused on machine learning-based methods, where we could successfully avoid domain and language-specific limitations. There are two types of machine learning-based approaches: supervised and unsupervised. A tagged corpus is used by supervised approaches to train the model. For each word in this section, a set of features is extracted. The characteristics could be either morphological, contextual, or word-level. When adjusting the model parameters, the tags of words serve as supervisors. In the absence of labelled data, NER can be performed using unsupervised approaches. In the case of unsupervised learning, there is no supervisor. From the data, they attempt to learn representations. Later, entity recognition can be done using these representations. Machine learning-based approaches, in contrast to rule-based approaches, are simple to adapt to new domains. In hybrid technologies, machine learning and rule-based algorithms are combined. The study of named entity recognition in Malayalam is not brand new. However, there are currently only a small number of reports. The majority of the works present in NER were created in western languages. In terms of NER, Indian languages lag considerably behind European languages. The majority of works in Indian languages have been recorded in Tamil and Hindi. Bindu M. S. reported the first Malayalam work in 2011 [1]. For NER, she employed a hybrid methodology. Her system makes use of both language ideas and statistical techniques. Jisha P. Jayan reported the second piece of work in 2013 [2]. For the NER assignment, she used TnT, an open-source statistical tagger. The training corpus, however, was incredibly limited. For training, just 10,000 words are used. Amritha University Coimbatore has released the third work as a part of FIRE-2014 [3]. They conducted a comparison study of entity tagging methods in various languages.

Table 1. Named Entity Tags and descriptions

Tag	Description
B-PER	Beginning of Person
I-PER	Inside Person
B-LOC	Beginning of Location
I-LOC	Inside Location
B-ORG	Beginning of Organization
I-ORG	Inside Organization
MISC	Miscellaneous entities
PUNC	Sentence and marker
OUT	None of the above tags

To evaluate NER in English, CRF is employed, while SVM is used for other languages. In contrast to other common training data sets, the training corpus size was likewise tiny. In 2016 [4], Shruthi S. created a NER for Malayalam. She applied TnT and MEMM in conjunction with a training corpus of 1230 sentences. Remmiya Devi et al. reported the most recent work (to our knowledge) on the NER system for Malayalam in 2016 [5]. Using structured word embedding based on skip-grammes, they attempted to extract named things from social media material.

Table 2. Features and descriptions

Feature	Descriptions
W	Word embedding of the target word
W-1	Word embedding of the preceding word
W-2	Word embedding of the second preceding word
POS	POS of the target word
Suffix	Suffix embedding of the target word
POS-1	POS of the preceding word
POS-2	POS of the second preceding word

3. Proposed work

We offer a NER methodology that makes use of neural networks. The input layer, hidden layer, and output layer are the three layers that define neural networks. The input layer receives the features that were retrieved from the training data. Before using an activation function to introduce nonlinearity in the hidden layer, the input layer features are multiplied by a set of weights. To get to the output layer, which produces the prediction, the output from the hidden layer is multiplied by a set of weights. A cost function is used to determine the difference between the true value and the forecasted value. To reduce the discrepancy between the anticipated value and the actual value, the *backpropagation algorithm* is used. With respect to network parameters, the backpropagation method determines the derivative of the cost function. Only when the weights for the output layer have been minimized are the hidden layer weights changed. Until the discrepancy between the actual value and the predicted value is acceptable, this process is repeated. Natural language text processing greatly benefits from the use of neural networks. They have several concealed layers [6]. Hierarchical feature learning is the defining property of these layers. Neural networks' hidden layers facilitate the learning of higher-level features from lower-level data. Figure 1 depicts the proposed system's design. For training, preprocessed, labelled text is employed. A set of characteristics corresponding to each word in the training corpus replaces the original term. The target word's word embedding, POS, suffix information, POS of the preceding words, and word embedding of the preceding words are among these aspects.

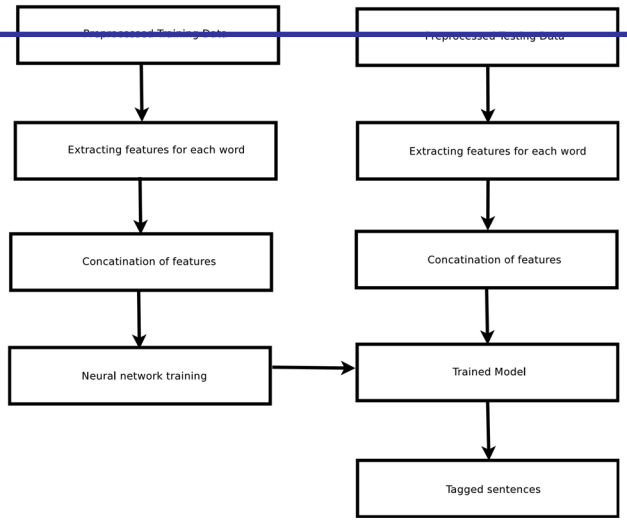

Fig. 1. Architecture of the proposed system

Table 2 displays each feature along with a brief description. The feature set for each word is chosen under the presumption that the contextual and morphological properties of the target word together provide the entity information for that word. Therefore, we made the decision to test a set of seven features. We used word embeddings to represent words and suffixes because they cannot be directly fed to neural networks. Word representations in a semantic space are known as word embeddings" [7]. Comparable meanings will result in comparable vectors for words. Words are transformed into vectors using Word2Vec. In order to conduct studies, word vectors of various sizes are constructed. Word2Vec also converts the suffix information of words into vectors. Words' POS information is combined into a single hot vector [8]. As the BIS tagset is used to tag words, the length of the one hot vector will be 36. The POS data from the words that came before is likewise combined into a single hot vector. A single vector that represents each word is created by concatenating all the features. To ensure that the feature vectors are all the same length, the necessary dummy vectors are entered into the feature set for the first two words in each phrase. To make the training process easier, the labels for each word are combined into a single hot vector. The trained neural network is then given the processed data.

The POS-tagged text from the corpus is used as the testing data throughout the testing phase. Similar to the training phase, features matching each word in the test data are retrieved. The trained model is then fed the concatenated set of features for predicting entity tags. The standard evaluation metric of accuracy is used to evaluate the anticipated tags.

4. Experiments and Results

For Malayalam NER, there is a shortage of available tagged corpus. Therefore, we ran our experiments using Kaggle's POS tagged corpus[9]. Approximately 6500 words from the corpus have named entity tags manually applied. The manual tagging phase employs the BIO(Begin, Inside, Outside) labelling scheme[10]. For tagging, only the top four classes of entities are taken into account. They fall under the categories of person, place, organization, and other classes. A particular tag's B-prefix designates the start of a specific named entity, and its I-prefix designates its interior. Always follow the B-tag is the I-tag. Table 1 lists many tags along with their descriptions. The dataset is split between 80% training sets and 20% validation sets following the manual tagging process.

Table 3. Performance of the system for different features

Feature set	Accuracy
Word	85.4%
Word, POS	86.9%
Word, POS, Suffix	91.3%
Word, POS, POS-1, Suffix	91.45%
Word, POS, Word-1, Suffix	94.4%
Word, POS, Word-1, Word-2, Suffix	95.3%
Word, POS, POS-1, POS-2, Word-1, Suffix	94.25%
Word, POS, POS-1, POS-2, Word-1, Word-2, Suffix	95.33%

A 6500 word corpus that was manually constructed is the foundation for the Word2Vec model. Word2Vec[11] is produced using a skip-gram arrangement. The minimum count is set to one, and the context window size is set to 10. Vector sizes vary depending on the model being built. The Word2Vec online training tool deals with terms that are not in your vocabulary. Using Word2Vec, suffix information about words is also embedded. The suffix embedding's size is set to 100. Every word in the corpus is given to a suffix stripper, which looks for the largest suffix that matches the set of rules it has stored. The matching suffix itself is returned if there is a match. If not, a five-character suffix is returned. As a result, every word in the corpus is replaced by a suffix from the stored rules or a five-letter version of that word's own suffix. The corpus is then subjected to Word2Vec. For this, 340 suffix stripping rules have been created [12]. The guidelines for suffix stripping have been developed for the Malayalam text's most frequent suffixes.

Tensorflow is used in the network's backend and is implemented in Python [13]. There are 25 training epochs. The loss increases as the number of epochs increases beyond 25. Therefore, we chose to end our training after 25 epochs.

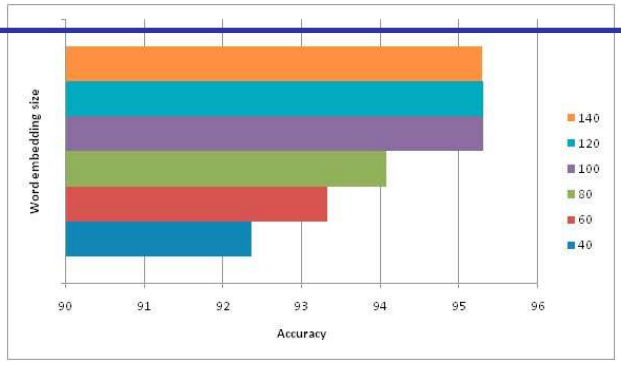

Fig. 2. Performance of the system for different word embedding sizes

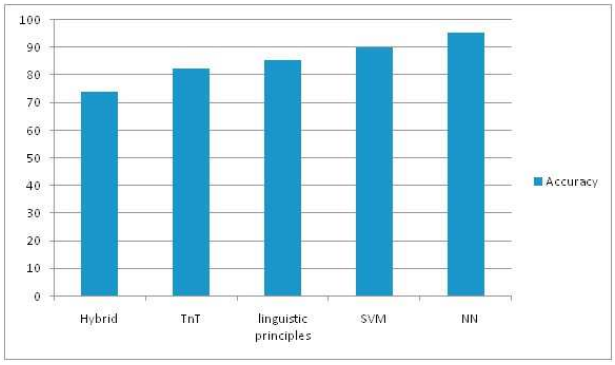

Fig. 3. Performance comparison with existing methodologies

The performance of neural networks is significantly impacted by the word vector size. We therefore made the decision to test word embeddings of various sizes. With various word embedding sizes, we got different results. Beyond 100, there was no gradual improvement in accuracy. As a result, the word embedding size is set at 100. Figure.2 displays the model's performance for various word embedding sizes. The choice of features is crucial in deciding how well the system performs. The system's performance for various feature combinations has been tested iteratively. Table 3 displays the impact of various feature combinations on our neural network. We empirically discovered that the quantity of training data significantly affects the NER system's performance.

The results of the studies demonstrate that our system works better than every other Malayalam NER approach. The system's overall tagging performance is 95.3%. The model's performance has been tested using various network parameters. We also ran 10-fold cross-validation on the data to increase the validity of the experiments. Figure.3 illustrates a comparison between the suggested system (NN) and various accepted approaches.

5. Conclusion

In this paper, we have discussed a NER system for Malayalam that is built on neural networks. This NER system's performance in entity recognition is its unique attribute. Our system outperforms all other methods for naming entities. More precise classification results from representing words as vectors. Suffix embedding appears to be necessary for inflectionally rich languages like Malayalam and may enhance efficiency to a certain extent. The majority of the applications covered in the literature rely on linguistic and statistical principles. For the purpose of recognizing named entities, none of them employ neural networks. According to the tagset we gave, the proposed system's total NER accuracy is 95.3%. By expanding the training data size, performance can be enhanced. In comparison to cutting-edge systems, our system performs better despite having fewer features. The suggested approach can also be applied to several NLP tasks, including speech recognition, phrase chunking, and POS tagging.

References

[1] MS Bindu and Sumam Mary Idicula. Named entity identifier for Malayalam using linguistic principles employing statistical methods. *International Journal of Computer Science Issues(IJCSI)*, 8(5), 2011.

[2] Jayan P Jisha, RR Rajeev, and Elizabeth Sherly. A hybrid statistical approach for named entity recognition for Malayalam language.

[3] M Anand Kumar Abinaya N, Neethu John and Soman KP. Amrita cen@ fire 2014: Named entity recognition for indian languages.

[4] Mr Jiljo and Mr Pranav PV. A study on named entity recognition for Malayalam language using tnt tagger & maximum entropy markov model. *International Journal of Applied Engineering Research*, 11(8):5425–5429, 2016.

[5] G Remmiya Devi, PV Veena, M Anand Kumar, and KP Soman. Entity extraction for Malayalam social media text using structured skip-gram based embedding features from unlabeled data. *Procedia Computer Science*, 93:547–553, 2016.

[6] Daniel C. Dennett. Introduction to deep neural networks, 2017. [Online; accessed 14-January-2018].

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] Tagged Malayalam corpus https://www.kaggle.com/datasets/disisbig/malyalam-news-dataset

[10] Wikipedia contributors. Insideoutsidebeginning (tagging) — Wikipedia, the free encyclopedia, 2017. [Online; accessed 14-April-2018].

[11] R Rehurek and P Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[12] RR Rajeev and Elizabeth Sherly. A suffix stripping based morph analyser for Malayalam language. In *Proceedings of 20th Kerala Science Congress*, pages 482–484, 2007

[13] Mart´ın Abadi and Ashish Agarwal. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.