raiden11@IECSIL-FIRE-2018 : Named Entity Recognition For Indian Languages

Ayush Gupta[1], Meghna Ayyar[2], Ashutosh Kumar Singh[3], and Rajiv Ratn Shah[4]

[1] Delhi Technological University, Delhi 110042, INDIA
gupta.ayush11786@gmail.com
[2] Indraprastha Institute of Information Technology, Delhi 110020, INDIA
leomi7ayyar@gmail.com
[3] Delhi Technological University, Delhi 110042, INDIA
ashu0788@gmail.com
[4] Indraprastha Institute of Information Technology, Delhi 110020, INDIA
rajivratn@iiitd.ac.in

Abstract. This paper presents our solution for the Named Entity Recognition (NER) task for the Information Extractor for Conversational Systems in Indian Languages challenge (IECSIL) [5] of the FIRE 2018 conference. A subset of the Information Extraction (IE) task, NER is a key to extract information and semantics of the text from unstructured data. The objective of NER is the identification and classification of every word or token in a document into predefined categories such as names of person, location, organization, etc. For this challenge the dataset provided by IECSIL [4] comprised of multilingual text of various Indian languages like Hindi, Tamil, Malayalam, Telugu, and Kannada. We mainly focus on the identification and classification of named entities belonging to nine categories like Name, Location, Datenum, etc. We tried linear models like Naive Bayes and SVM, and also a simple Neural Network to solve this problem. The best results are achieved by the simple neural network with an accuracy of 90.33% for all languages combined. This indicates that different advanced neural networks could be possible solutions to further improve this accuracy.

Keywords: Named Entity Recognition · Information Extraction · Word Embeddings · Neural Networks.

1 Introduction

The sheer volume of unstructured data available on the Web, in the form of blogs, articles, emails, social media posts, documents et cetera is so large that the task of deriving information from them demands an approach that does not involve the manual annotation of this data. The data is heterogeneous, which makes annotation by a human nearly impossible. Therefore, we would like to develop a computer annotation approach to structure the data and make the information from it easily extractable. Information extraction from unstructured data and text is an NLP technique that deals with this problem and entity recognition is one small subtask in the direction of solving it.

NER tasks traditionally require large amounts of knowledge in the form of feature engineering and lexicons to achieve high performance. Moreover, named entities are open class expressions that have a lot of varieties, where new expressions are being added constantly. NER deals with the location and identification of the named entities that are present in the sentences given in the textual input. It generally has pre-defined categories like names of people, places, organizations, terms specific to a particular topic or field, industrial product names et cetera which together comprise of the named entities [18]. NER is essential for various NLP tasks like Question Answering Systems [17], Information Retrieval [22], Machine Translation [3] etc. In addition to this, NER finds many applications in multiple industries such as news and media, search engines, content recommendations, customer support and also in academia.

Several techniques have been developed to recognize named entities for the English language and also for other foreign languages like Chinese, Japanese, Korean, Arabic, and Spanish. Many of these techniques use either rule-based technique shown by Kim and Woodland[13] or some form of statistical technique e.g. Malouf, Robert[16] used Hidden Markov Model to perform entity recognition. Both these approaches rely on the help of a language expert for the creation and validation of a large dataset which can then be used for further analysis. In contrast, for Indian languages, not much work has been reported because of insufficient resources due to which it has been difficult to employ statistical techniques for Indian languages. Moreover, due to the morphological nature of the Indian languages, it needs different methods than what has been employed for English to form a language model. For e.g. in the sentence *Peter likes Paris.*, NER needs to tag the entities *Peter* as <Name of a person> and *Paris* as <Name of place>. Here one rule to identify the nouns would be that nouns generally start with a capital letter. However, for Indian languages like Hindi, Tamil et cetera the nouns have a non-capitalized form and also incorporate a richer morphology as compared to English making it more challenging to perform NER [19].

The corpora provided as a part of the challenge consisted of pre-processed data for five Indian languages thereby, reducing the main difficulty of getting a validated and annotated dataset for the NER task. As an effort to perform entity recognition we have used some linear models as baselines, and have also used a neural network model to demonstrate that these techniques can be extended for the purpose of NER of Indian languages. The rest of the paper is organized as follows. The related work is presented in Section 2 while Section 3 describes the methodology followed in detail. Section 4 discusses the evaluation of the various models tried for the different languages in the corpora and Section 5 concludes the paper and presents some future work that can be pursued.

2 Related Work

Saha et al. [21] developed *A Hybrid Feature Set based Maximum Entropy (MaxEnt) Hindi Named Entity Recognition.* The four identified named entity (NE) by them were Person names (P), Location names (L), Organization names (O) and Date (D). MaxEnt, a supervised machine learning technique as given by Birthwick and Grishman [6] was applied to solve linguistic problems with the help of orthographic, collocation features and gazetteers lists. They used a 2-phase transliteration model to construct about 34 entities and subsequently employed a semi-automatic induction of context patterns used for classification. A 0.76 F-measure was achieved as a baseline result and 0.81 F-measure was achieved after adding gazetteer lists and context patterns into MaxEnt based NER system.

Saha et al. [20] implemented *Hybrid Approach for Named Entity Recognition in Indian Languages* for 5 Indian languages - Hindi, Bengali, Oriya, Telugu, and Urdu. They used two approaches i.e. the Linguistic approach where the typical rules were written by linguists and the Machine Learning (ML) approach – in which the system was trained using tags. They received poor accuracy for Oriya, Telugu and Urdu languages compared to the other two languages due to lack of Parts Of Speech (POS) information, morphological information, language-specific rules, and gazetteers lists.

CRF was introduced by Lafferty et al. [14] to build a probabilistic model for segmenting and labelling the sequence data. Ekbal et al. [7] have worked on *Named Entity Recognition in Bengali: A Conditional Random Field Approach* by using a CRF based approach to develop NER for Bengali. They achieved F-measure of 0.91 using CRF, more than the HMM result.

Gali et al. [8] discussed the ambiguities for Indian languages that deal with the linguistic issues like agglutinative nature and absence of capitalization, same meaning for common name and proper name, spelling variation, patterns, and suffixes. Using CRF's they performed statistical tagging, resolved capitalization etc. They achieved F - measures 0.41, 0.50, 0.39, 0.40, and 0.43 for Bengali, Hindi, Oriya, Telugu, and Urdu respectively.

Amarappa and Sathyanarayana [1] worked on *Named Entity Recognition and Classification (NERC) in Kannada language*, built a SEMI-Automatic Statistical Machine Learning NLP model based on noun taggers using HMM which was a challenging task. In addition, they extended their work in Amarappa and Sathyanarayana [2] to use a Multinomial Naïve Bayes (MNB) Classifier discussed by Kibriya et al. [12] to develop a novel model for NERC.

Malarkodi et al. [15] experimented with NER on Tamil database, which coped with real-time challenges using CRF's and could be applied for most of the Indian languages. Kaur and Gupta [11] built an NER for *Punjabi* language using rule-based and list look-up approaches. Punjabi is a language with high clung and inflections, which leads to linguistic problems. The rule-based approach trained the system to identify NE by writing rules manually for all NE features.

3 Methodology

The problem at hand is to use multilingual corpora to perform entity recognition task for five Indian languages. The following sections describe in detail the various steps that have been performed to achieve the results. Section 3.1 describes the system architecture followed by Section 3.2 which discusses feature extraction techniques applied. Finally, Section 3.3 deals with the description of the various models used in our approach.

3.1 System Architecture

The complete architecture of the proposed system has been summarized in Fig. 1. The pipeline of the system starts with the given dataset, as the input. We marked all the full-stops and punctuation marks as entities belonging to Other class and removed them. Then, we marked the numerical entities as all of them belong to Numbers or Datenum class. This reduced the dataset size and aided in feature extraction as we do not have word embeddings for numerical entities. As different languages can have different ways of writing digits, we have to modify the digit recognizer for each new language we add. This step is followed by the process of feature extraction which has been described in detail in Section 3.2. Once the features were extracted as vectors, we experimented with various models like Naive Bayes and SVM, selected the one which gave the best results in terms of accuracy by predicting the labels on unseen data. For each entity, we selected the class having the highest probability among all.

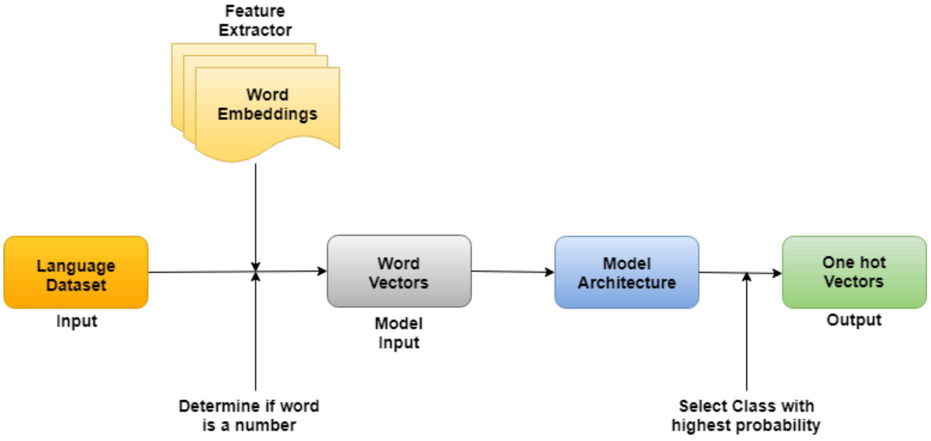

Fig. 1. System Architecture Diagram

3.2 Feature Extraction

An effective representation for sentences is to use the Bag of Words (BoW) model which can then be given as input to a linear classifier like Naive Bayes or Support Vector Machine (SVM). However just using BoW poses problems like large feature dimension and sparse vector representation to represent words. Word embeddings can capture the semantic relations amongst the words, unlike BoW. Hence, we used FastText, a fast word embedding/vector generator by Facebook [10] that provides a way to represent words using numbers. We used a pre-trained collection[5] of word vectors in 294 languages. These word vectors were trained using skip-gram model described by Bojanowski et al. [9] with default parameters. Each word vector consists of 300 decimal numbers representing unique features. Words which are semantically close in meaning appear together in the 300-dimensional graph. Word vectors were downloaded for all five Indian Languages and stored as binary files. Except for Number and Datenum, all other classes, as shown in Table 1, have more than 90% of their entities present in these vectors. Overall 95.83 % of the entities were present. Remaining entities were marked as unknown and a randomly generated sequence of decimals between -1 and 1 was used as their feature sets.

Table 1. Presence percent of training set entities in Facebook word embeddings (Org: Organization)

Language	**Event**	**Things**	**Org**	**Occupation**	**Name**	**Location**	**Other**	**Average Presence**
Hindi	99.69	99.33	99.23	99.48	94.96	98.91	96.38	98.28
Kannada	98.85	97.11	96.85	96.92	89.17	96.94	89.4	95.03
Malayalam	94.86	96.65	97.17	95.72	90.71	96.52	86.14	93.96
Tamil	98.34	98.3	97.95	96.93	91.72	95.13	93.05	95.91
Telugu	98.9	99.16	98.72	98.72	83.65	99.15	93.48	95.96
Class Avg.	98.12	98.11	98.00	97.55	90.04	97.33	91.69	**95.83**

3.3 Model Description

A simple and efficient baseline for our model is to use a linear classifier, e.g. a logistic regression or a SVM algorithm. However, these classifiers fail to capture the context of the statement while evaluating entity. Common examples are the words; *book, battle, address, bear etc.*, which can represent a verb as well as a noun depending on the context. Also, they cannot generalize well in case of imbalance in the class distribution in the dataset. To overcome this, we can apply a multi-layer neural network or a deep neural network which can capture the composite relation between the words.

[5] https://github.com/facebookresearch/fastText/blob/master/pre-trained-vectors.md

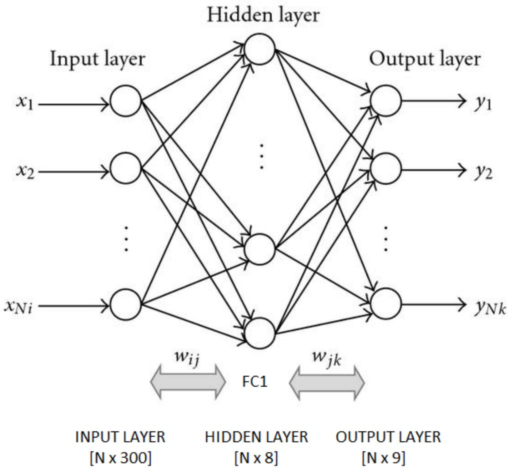

Fig. 2. Artificial Neural Network model diagram

Neural network algorithms tend to exploit patterns and structures in datasets to discover useful representations. So we chose to build a simple artificial neural network, with one fully connected hidden layer with 8 units using the TensorFlow library as shown in Fig. 2. The input layer consists of 300 nodes without any activation function while the following layers have a softmax activation function. The following equation describes the forward propagation in the first hidden layer of the neural network :

$$a_i^{l+1} = g\left(\sum_{j=1}^{300} \left(a_j^l * W_{ij}\right) + b^l\right) \tag{1}$$

Here a_i^{l+1} is the activation value of the ith node at the $(l+1)$th layer, W_{ij} is the weight from jth node of the lth layer to the ith node, b_l is the bias at the lth layer and g is the softmax activation function. In our case, we have used a basic neural network with one input layer of 300 units, one fully connected layer of size 8 followed by the output layer of size 9 units. So $l = 0$ for the first hidden layer.

Finally, the output layer takes the values from the hidden layer nodes as input, adds a bias value and returns a one hot vector as output which represents the predicted class of the entity. The number of training epochs was fixed to 20 as above this number we could not observe any improvement in the performance of the model in terms of accuracy. It takes about 2 hours to execute the model on each language.

4 Evaluation

Section 4.1 describes the data while Section 4.2 provides insight into the experimental setup used. Section 4.3 lists the evaluation metrics used. Section 4.4 presents the results obtained in detail while Section 4.5 presents Error Analysis.

4.1 Dataset Description

The corpora consisted of five Indian languages namely, Hindi, Kannada, Malayalam, Tamil, and Telugu. It was provided by ARNEKT Organization[6] for the purpose of this challenge. The dataset comprised of files for each language in which the sentences were separated by *newline string*. Most of the sentences provided were about 20 to 35 words long although some extended to above 100 words as well. The details about the entity categories and their distribution in the different languages are shown in Table 2. The entities were tagged into 9 separate classes namely number, event, Datenum, things, organization, occupation, name, location and other.

Table 2. Description of Dataset (Num: Numbers, Org: Organisation, Occ: Occupation, Loc: Location.)

Languages	Num	Event	Date/ Num	Things	Org	Occ	Name	Loc	Other
Hindi	37754	2932	2672	4048	12254	15732	88887	166547	1141207
Kannada	3948	523	1046	242	746	3081	15110	10473	262651
Malayalam	30553	837	1609	1999	4841	8037	59422	29371	701664
Tamil	77310	5112	15482	6183	9999	16507	118021	134262	1109354
Telugu	28618	729	2521	1069	2431	8437	60499	95756	577625

4.2 Experimental Setup

The dataset provided was properly annotated and hence did not need any kind of pre-processing except the removal of punctuation marks. By using the words vectors provided by Facebook, these words were then vectorized to a form that could be directly given as input for our models. The supervised dataset was divided into training and testing sets in the ratio 8:1.

[6] http://iecsil.arnekt.com

4.3 Evaluation Metrics

Accuracy has been used as the metric for evaluating classification models in our experiments. Accuracy is defined as the ratio between a total number of correct predictions $\mathrm{N_c}$ upon the total number of predictions $\mathrm{N_t}$.

$$\text{Accuracy} = \frac{N_{\mathrm{c}}}{N_{\mathrm{t}}} \tag{2}$$

In addition to classification accuracy we have also used :

1. Recall (R) : Calculates the fraction of entities identified by the model and the total number of named entities actually present in the corpus. For instance let $\mathrm{N_t}$ be the total number of named entities present in the corpus and $\mathrm{N_i}$ be the number of entities identified by the model.

$$\text{Recall} = \frac{N_{\mathrm{i}}}{N_{\mathrm{t}}} \tag{3}$$

2. Precision (P): Calculates the fraction of the named entities that were correctly identified by the model. For instance let $\mathrm{N_i}$ be the total number of named entities actually present in the corpus that were correctly identified and $\mathrm{N_e}$ be the total number of named entities identified by the model.

$$\text{Precision} = \frac{N_{\mathrm{p}}}{N_{\mathrm{e}}} \tag{4}$$

3. F1-Score: F1 score is the harmonic mean of precision and recall. A perfect F1 score of 1 is reached when the model has perfect precision and recall and worst at 0.

$$\text{F1-Score} = 2*\frac{\mathrm{R} * \mathrm{P}}{\mathrm{R} + \mathrm{P}} \tag{5}$$

4.4 Results and Analysis

The model was initially tested on the basic implementations of Naive Bayes Classifier and Support Vector Machine(SVM). They yielded an overall accuracy of 81.42% and 86.50% respectively. The results for all the languages provided in the dataset have been listed in Table 3. It is evident from the reported figures that the artificial neural network performs better than the standard SVM and Naive Bayes models. We observe that the accuracy values of the neural network lies in the vicinity of 90% and are relatively close to each other. This is because our model is language independent and thus, it should yield similar values for any new language. We can also observe that Tamil has comparatively lower accuracy than other languages. This could be due to large word vectors corpus for Tamil as compared to other languages.

The detailed classification results of all 5 languages for each of the 9 classes on the FIRE 2018 dataset are presented in Table 4. Fig. 3 depicts a graph comparing recall values for all the languages. Similarly, Fig. 4 and Fig. 5 depict

Table 3. Accuracy results for entity recognition for various models

Language	ANN	SVM	Naive Bayes
Hindi	91.52	87.20	78.95
Kannada	**92.14**	**88.15**	**86.62**
Malayalam	90.27	86.80	84.40
Tamil	87.72	83.48	79.27
Telugu	90.02	86.87	77.90
Average	**90.33**	**86.50**	**81.42**

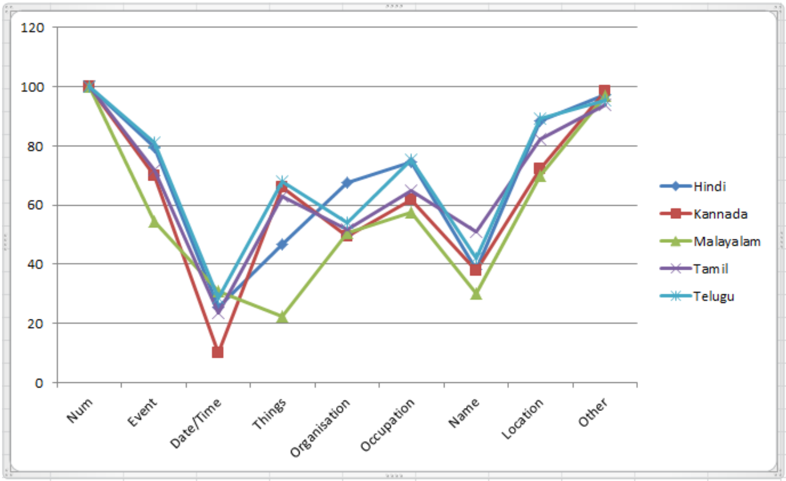

Fig. 3. Class wise Recall Values

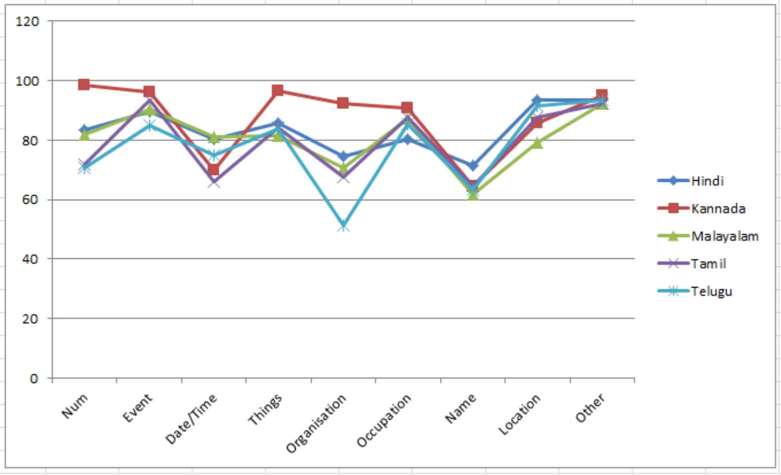

Fig. 4. Class wise Precision Values

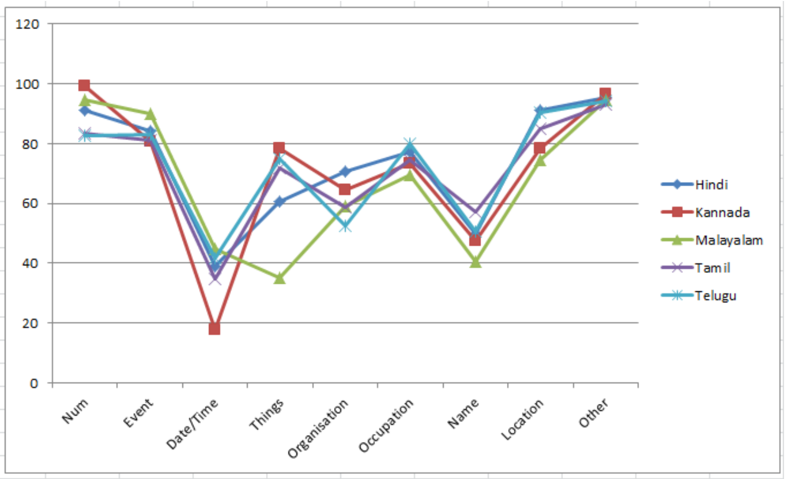

Fig. 5. Class wise F1-Score

graphs comparing precision values and F1-Score respectively. The recall rate of Number class is 100% for all the languages due to the digit recognizer. Many Datenum entities get tagged as Number entities thereby reducing the precision value of Number class and recall value of Datenum class. This points to the need to adopt a different strategy to recognize Datenum class.

Table 4. Class wise metrics for entity recognition for ANN model

		Num	**Event**	**Date/ Num**	**Things**	**Org**	**Occ**	**Name**	**Loc**	**Other**
Hindi	Recall	**100.0**	79.48	25.65	46.80	67.40	74.38	38.15	88.47	97.21
	Precision	83.62	89.71	80.40	85.77	74.41	80.39	71.27	**93.61**	93.33
	F1-Score	91.08	84.29	38.91	60.56	70.73	77.27	49.70	90.97	**95.23**
Kannada	Recall	**100.0**	69.73	10.11	66.11	49.32	61.63	37.72	72.04	98.46
	Precision	**98.56**	96.29	70.00	96.38	92.46	90.64	64.69	85.73	94.83
	F1-Score	**99.27**	80.88	17.66	78.43	64.33	73.37	47.68	78.29	96.61
Malayalam	Recall	**100.0**	54.65	30.88	22.47	50.72	57.57	30.00	69.98	96.91
	Precision	81.88	90.38	81.25	81.66	70.72	87.24	61.99	79.37	**92.35**
	F1-Score	**94.58**	90.03	44.80	35.25	59.07	69.37	40.43	74.38	94.58
Tamil	Recall	**100.0**	71.91	23.45	63.02	51.60	64.79	51.03	82.17	93.94
	Precision	71.70	**93.51**	65.88	84.05	67.68	87.84	64.66	87.58	92.49
	F1-Score	83.58	81.29	34.71	72.03	58.60	74.57	57.04	84.78	**93.20**
Telugu	Recall	**100.0**	81.15	28.70	67.93	53.96	75.34	41.91	89.34	95.28
	Precision	70.51	84.84	74.80	83.96	51.32	85.37	63.29	91.72	**93.32**
	F1-Score	82.71	82.96	41.68	75.10	52.61	80.04	50.43	90.51	**94.29**

4.5 Error Analysis

We did an error analysis to identify the incorrect predictions. In all the languages, several Name entities were misclassified as Other entities. This is because Name entities have relatively lower presence in the Word Embeddings leading to a random vector getting assigned to represent them during the feature extraction stage. One way to solve this would be the addition of a language-specific gazetteers list to separate out the Names beforehand. The model also misclassifies Things and Organization entities at times. This is due to the diverse nature of these entities and also because they are rarely repeated in the dataset. The classifier also experiences some problems while differentiating between Location and Name entities. Along with Number entities, Event and Occupation entities are also classified with high accuracy and very low misclassifications. This is because of the limited set of common Events and Occupations entities present in both the real world and consequently in the dataset.

5 Conclusion and Future Work

The increasing amount of unstructured data requires that we develop an approach to extract useful information in an efficient way. Information Extraction (IE) from data is an important area of research of which named entity recognition is a subset. In this paper, we have described the approach we adopted for developing models capable of performing the task of NER for IECSIL as a part of the FIRE 2018 challenge. The exact methods followed have been discussed in detail. We have provided the results obtained along with analysis about the entities for which the models performed well and reasons for such a behaviour. The highest accuracy for NER was 90.33% for all languages combined when we used our simple ANN model, which is an improvement to the baseline accuracy of 85.73%.

As an extension to the model presented, we can work on deeper models with a better-suited structure for handling textual data. Recurrent Neural Network (RNN), sometimes even addressed as Long short-term Memory (LSTM) is one such example. A further improvement to this model is RNN with added attention mechanism as demonstrated in Zhou, Peng et.al.[23], where they have shown that RNNs with attention mechanism perform well on NLP related problems. In addition to this, we can work towards adding language specific markers and special features like adding a Datenum recognizer that can reduce the misclassifications, enhance the model and increase its robustness.

References

1. Amarappa, S., Sathyanarayana, S.: Named entity recognition and classification in kannada language. International Journal of Electronics and Computer Science Engineering **2**(1), 281–289 (2013)

2. Amarappa, S., Sathyanarayana, S.: Kannada named entity recognition and classification (nerc) based on multinomial na\" ive bayes (mnb) classifier. arXiv preprint arXiv:1509.04385 (2015)
3. Babych, B., Hartley, A.: Improving machine translation quality with automatic named entity recognition. In: Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT. pp. 1–8. Association for Computational Linguistics (2003)
4. Barathi Ganesh, H.B., Soman, K.P., Reshma, U., Mandar, K., Prachi, M., Gouri, K., Anitha, K., Anand Kumar, M.: Information extraction for conversational systems in indian languages - arnekt iecsil. In: Forum for Information Retrieval Evaluation (2018)
5. Barathi Ganesh, H.B., Soman, K.P., Reshma, U., Mandar, K., Prachi, M., Gouri, K., Anitha, K., Anand Kumar, M.: Overview of arnekt iecsil at fire-2018 track on information extraction for conversational systems in indian languages. In: FIRE (Working Notes) (2018)
6. Borthwick, A., Grishman, R.: A maximum entropy approach to named entity recognition. Ph.D. thesis, Citeseer (1999)
7. Ekbal, A., Haque, R., Bandyopadhyay, S.: Named entity recognition in bengali: A conditional random field approach. In: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II (2008)
8. Gali, K., Surana, H., Vaidya, A., Shishtla, P., Sharma, D.M.: Aggregating machine learning and rule based heuristics for named entity recognition. In: Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages (2008)
9. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893 (2018)
10. Grave, E., Mikolov, T., Joulin, A., Bojanowski, P.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL. pp. 3–7 (2017)
11. Kaur, K., Gupta, V.: Name entity recognition for punjabi language. Machine translation **2**(3) (2012)
12. Kibriya, A.M., Frank, E., Pfahringer, B., Holmes, G.: Multinomial naive bayes for text categorization revisited. In: Australasian Joint Conference on Artificial Intelligence. pp. 488–499. Springer (2004)
13. Kim, J.H., Woodland, P.C.: A rule-based named entity recognition system for speech input. In: Sixth International Conference on Spoken Language Processing (2000)
14. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
15. Malarkodi, C., Pattabhi, R., Sobha, L.D.: Tamil ner–coping with real time challenges. In: 24th International Conference on Computational Linguistics. p. 23 (2012)
16. Malouf, R.: Markov models for language-independent named entity recognition, proceedings of the 6th conference on natural language learning. August **31**, 1–4 (2002)
17. Mollá, D., Van Zaanen, M., Smith, D., et al.: Named entity recognition for question answering (2006)
18. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)

19. Pillai, A.S., Sobha, L.: Named entity recognition for indian languages: A survey. International Journal **3**(11) (2013)
20. Saha, S.K., Chatterji, S., Dandapat, S., Sarkar, S., Mitra, P.: A hybrid approach for named entity recognition in indian languages. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian languages. pp. 17–24 (2008)
21. Saha, S.K., Sarkar, S., Mitra, P.: A hybrid feature set based maximum entropy hindi named entity recognition. In: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I (2008)
22. Salomonsson, A.: Entity-based information retrieval. Department of Computer Science, Faculty of Engineering, LTH, Lund University (2012)
23. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 207–212 (2016)