Named Entity Recognition for history data in Kannada Language

Harshitha Parameshwara Rukmini	hparmes@usc.edu	6186506854
Komal Nagpal	knagpal@usc.edu	3639028468
Nagarjun Gopalaswamy	ngopalas@usc.edu	1170544051
Sruthi Madapoosi Ravi	madapoos@usc.edu	5196555971

1 INTRODUCTION

This report presents the development of NER system for Kannada language using statistical Conditional Random Fields and Neural Networks.

Kannada is a highly inflectional and an agglutinating language providing one of the most challenging sets of linguistic and statistical features. The language uses 49 phonemic letters. The number of written symbols, however, is far more than the 49 characters in the alphabet as different characters can be combined to form compound characters (ottakshara). Each written symbol in the Kannada script corresponds to one syllable, as opposed to one phoneme in languages like English. The Kannada script is syllabic.

Kannada, like other Indian languages, is a resource-poor language – annotated corpora, name dictionaries, good morphological analyzers, Parts of Speech (POS) taggers etc. are not yet available in the required measure. Inadequacy of applications to identify the namworded entities for Kannada language inspired us to build a Named Entity Recognition System. This would serve as a primary step to develop advanced applications like Chat bots and Question answering systems.

1.1 Previous work

Previous work in Kannada NER include classification using Multinomial Naive Bayes (MNB) and HMM based NERC by S Amarappa et al [3]. The most recent work in Kannada NER using CRF is still in embryonic stages using limited features [4].

Due to the lack of large tagged corpora, it is difficult t build a NER system using state-of-the-art techniques such CRF and Neural Networks. This motivated us to make novel contribution to the Kannada NLP community by building a reliable NER system using history domain corpora. Our significant contributions include creating a full-fledged NER annotated corpora and predicting NER tags using CRF and Neural Networks.

We developed a CRF based NER system using well defined features like POS tags, morphemes and contextual features. We achieved an accuracy of 74.16% and we implemented NER using Neural Networks which automatically learns features from the data and resulted in accuracy of 73.54%. We will open source the word vectors and the trained Neural Net model to the Kannada NLP community for future research.

2 METHOD

2.1 Materials

2.1.1 Corpus

The first step is to generate tagged corpora and we collected the data from the following sources:

1. History Textbooks in Kannada.[14]
2. EMILLE (Enabling Minority Language Engineering) untagged corpus.[13]
3. Crawled data from Wikipedia.

We trained the models (CRF and NN) using **80%** of the training set and tested using **10%** validation set and **10%** test set. Training data comprised of 16K sentences (129500 tokens). validation and test set comprised of 16309 tokens each. Additionally, for Neural Networks we used crawled data from Wikipedia (8 Million tokens) to train word vectors.

2.1.2 Tools

We used the following open source tools available in Kannada language for language specific features-

1. POS tags annotated by an existing Kannada POS tagger (by Siva Reddy) [15]
2. A morphological analyzer to segment words (Indic NLP library:Anoop Kunchukuttan) [16]
3. CRF++ Toolkit for NER using tagged corpora.[17]

2.1.3 Tagging

Two of our team members manually tagged the corpus. We adapted the standard CoNLL 2003 guidelines for tagging 3 named entities: Person, Organization and Location. In addition, we have created 4 more named entities: Role, Community, Date and Number using the guidelines mentioned in [Table 1]. This data is converted to standard IOB format where every entity begins with a tag B-XXX followed by I-XXX if it is a multi-word named entity. The corpus contained 24% Named Entities and the rest were Not Named Entities.

2.1.4 Named Entities and Guidelines

Before the manual tagging process, in order to achieve best manual tagging consistency we outlined the tagging specification as mentioned in [Table 1].

Named Entity	**Description**	**Kannada Language**	**Transliterated**	**English Translation**
PER	Names of persons	ನರೇಂದ್ರ /B-PER ದಾಮೋದರದಾಸ್/I-PER ಮೋದಿ /I-PER	Narendra/B-PER Damodardas/I-PER Modi/I-PER	Narendra/B-PER Damodardas/I-PER Modi/I-PER
LOC	Places, monuments, rivers and mountains	ಹಿಮಾಚಲ /B-LOC ಪ್ರದೇಶ/I-LOC	Himachala/B-LOC Pradesha/I-LOC	Himachal/B-LOC Pradesh/I-LOC
COM	Kingdom, dynasty, empire or a community	ಚಾಲುಕ್ಯರು/B-COM	Chalukyaru/B-COM	Chalukyas/B-COM

DATE	Date or year or a mention of a period of time	15 ನೇ/B-DATE ಆಗಸ್ಟ್/B-DATE ೧೯೪೭/B-DATE	12ne/B-DATE August/B-DATE 1947/B-DATE	12th/B-DATE August/B-DATE 1947/B-DATE
NUM	Numerals or numbers represented in words	ನೂರು/B-NUM ಒಂದು/I-NUM	Nooru/B-NUM Ondhu/I-NUM	Hundred/B-NUM One/I-NUM
ORG	Organization, Government, department or committee	ಭಾರತೀಯ/B-ORG ವಿಜ್ಞಾನ/I-ORG ಕೇಂದ್ರ/I-ORG	Bharatiya/B-ORG Vijnana/I-ORG Kendra/I-ORG	Indian/B-ORG Institute/I-ORG Science/I-ORG
ROL	Roles/positions held by people	ಕೃಷಿ/B-ROL ಮಂತ್ರಿ/I-ROL	Krushi/B-ROL Manthri/I-ROL	Minister/B-ROL Agriculture/I-ROL
NNE	Not named entity	ಅಭಿವೃದ್ಧಿ/O ಶಾಂತಿ/O	Abivruddhi/O shanti/O	Development Peace/O

Table 1: Named Entity Tag Guidelines

2.1.5 Corpus Ambiguities

1. Spelling variations

Different word inflections were treated as separate words which caused ambiguities while training.

Example: In some places the word *Wodeyar* appeared as “ಒಡೆಯರ್”(*Odeyar*) and in other places as “ವೊಡೆಯರ್”(*Vodeyar*).

2. Noise in collected data

Noise in crawled wiki articles include spelling mistakes and punctuation inconsistencies.

Example: ಋಷಿ and ರುಷಿ are two different versions of representation for the word *sage*. Even though the correct representation is ಋಷಿ, but ರುಷಿ is accepted too.

3. Consistency of named entities

Same named entity appears in different word forms in the corpus. The below two variations are considered as two separate entities by the model.

ಮಾರ್ಚ್ ಮಾರ್ಚಿ
March Marchi
March March

2.2 PROCEDURE

The NER system is built using two approaches i.e Conditional Random Fields and Neural Networks.

2.2.1 Experimental Procedure for Conditional Random Fields

2.2.1.1 Preprocessing

1. Extracted data from history textbooks and Wikipedia articles using Java crawler4j library.
2. Manually tagged the data as per the standard CoNLL 2003 guidelines.
3. Tokenized the data and extracted the root and suffix of the tokens using Morfessor[16]. The accuracy of the Morfessor was 74% with manual evaluation on a random sample of 1000 words.
4. Extracted the POS tags using the POS tagger[15] which has accuracy of 87% as reported by the publisher. The accuracy of the POS tagger was 89% with manual evaluation on a random sample of 1000 words.
5. Consolidated the output of POS tagger, Morfessor along with the manually tagged NER data and created the input data for CRF++[17].

2.2.1.2 Training

We trained the CRF model and identified the best set of features with 129K tokens and tested with the validation set of 16K tokens. The final trained model includes the training and validation set and has 145K tokens. The resulting model is tested on 16K tokens of test data.

2.2.1.3 Feature Description

The following features have been used with the window size of [-2,2].

1. Word feature - Previous and next word of a particular word can be used to capture the context.
2. POS feature - POS of the current and the surrounding words was used as features
3. Root and Suffix feature - Language features such as root and suffix of the current and the surrounding words were used to identify named entities
4. Date and Digit feature - Binary date feature is created based on both the digit check (0-9) and transliterated Kannada digits (unicode character set). The month and days are identified from a predefined list configured in our system. The example below shows the numerals in Kannada recognized by the Date feature.

Numeral	Number	English
೦	ಸೊನ್ನೆ/Sonne	Zero
೧	ಒಂದು/Ondhu	One

5. NER feature - NE tag of previous word was considered as feature
6. Bigram features - A combination of root, suffix and POS tag was used as bigram feature

2.2.2 Experimental Procedure for Neural Networks

2.2.2.1 Word embedding Generation

As Kannada language is a morphologically rich language, word context alone is not sufficient to generate word embeddings. Instead, we used character enhanced word encoding[8] which includes both word vectors and composition of a word to generate embedding for a word. We created 100 dimensional word vectors with a vocabulary size of 134746 words. The data for training word embedding model has both history domain data and out of domain data.

2.2.2.2 Preprocessing

The train, validation and test set files are created which consists of words, character composition of a word and NER tag. The system is supplied with a *character vocabulary* which consists of all the unicode characters of Kannada language. A *tag vocabulary* file is created which consists of all the NER tags that the system should be trained to identify on the test set. A *pretrained vector file* is provided to speed up the learning phase. All the training, development and test files are created in a three column format where in the first column corresponds to a word, second column corresponds to characters separated by spaces and third column corresponds to the NER tag. We created a clean corpora consisting of only Kannada characters and numerals excluding any special characters for generating word vectors.

Example:

ಭಾರತ/Bharatha ಭ ಾ ರ ತ B-LOC

ಯುರೋಪಿನ/Yuropina ಯ ು ರ ೋ ಪ ಿ ನ B-LOC

2.2.2.3 Training

We implemented a model which encoded both character level as well as word level information for named entity recognition. It consists of 3 components[9]

1. Convolution layer for extracting character level features - Each word is represented as an array of character vectors and 15 filters are applied to filter 1 to 5 character sequences.
2. A bidirectional LSTM for encoding input word sequences - Input to the bidirectional LSTM is a sequence of words where each word is represented by the concatenation of character and word vectors.
3. A feed forward output layer for predicting tags - This is used as a decoder which takes input from the previous layer as well as the previous tag to predict tag for the current word.

2.3 Evaluation Criteria

2.3.1 Annotation

Two of our team members manually annotated the corpus and the inter-annotator agreement is expressed in terms of Krippendorff's alpha coefficient. We have observed a coefficient of 0.87 for the corpus. The annotators used the tagging guidelines [Table 1] to overcome the conflicts that arose during individual annotation. Initially, the annotation was done together on a random sample of 5000 words which helped to resolve the conflicts and update the guidelines. The rest of the corpus was tagged individually and the coefficient was calculated for the same. The inconsistency in inter-annotator accuracy was caused due to unfamiliar words, ambiguity between tokens, similar prefix of the words and due to human error. [Table 2] shows the few inconsistencies observed.

Word	Transliteration	Annotated Tags	Actual Tags
ಭಾರತ	Bhaaratha (India)	B-LOC	B-LOC

ಭಾರತೀಯರು	Bharathiyaru (Indian)	B-LOC	B-COM
ವಿಜಯನಗರ ಸಾಮ್ರಾಜ್ಯ	Vijayanagara(Vijayanagar) Samrajya(Dynasty)	B-LOC NNE	B-COM I-COM
ದಿವಾನ್ ನವಾಬ್	Dewan(Minister) Nawab(King)	B-PER B-COM	B-ROL B-ROL

Table 2: Ambiguities in Manual Tagging

2.3.2 NER Prediction

The trained NER system's performance for CRF and Neural Networks is measured in terms of *Precision (P), Recall (R) and F1-measure (F1).*

P = No. of correctly labeled named entities / Total No. of labeled named entities

R = No. of correctly labeled named entities / Total No. of named entities in data set

F1-Measure = 2PR / (P + R).

2.3.3 Baseline

We considered the CRF model developed by S Amarappa et al [4] with an of accuracy of 37% for the generic Kannada domain as our baseline system. We evaluated the baseline system with our history domain data by considering limited word features and got an accuracy of 56%.

3 RESULTS

Evaluation results of the CRF and NN based NER system for the test set in terms of overall F1-Score is specified. We have calculated the macro and micro F1-Score considering the factor that the data contains maximum number of not named entities.

3.1 Experimental Results for CRF

The table shows the results of CRF for different feature sets.

Feature Set ID	Features	F1- Score Macro-Average	F1-Score Micro-Average
1	Current Word (W), W-1, W-2 (1)	0.5697	0.8549
2	W, W-1, W-2,W-3 (16)	0.5643	0.8579
3	W, W-1, W+1 (11)	0.6023	0.8655
4	W, W-1, W-2, First Word (FW) (2)	0.5504	0.8615
5	W, W-1, W-2, W+1, W+2 (12)	0.5898	0.8646
6	W, W-1, W-2, NE tag of W (3)	0.6377	0.8817
7	W, W-1, W-2, W+1, NE tag of W (15)	0.6221	0.8812

8	W, W-1, W-2, Root Word of W, W-1 and W+1 Suffix of W, W-1 and W+1 NE tag of W (4)	0.6857	0.8992
9	Root Word of W, W-1 and W+1 POS Tag of W,W-1,W-2,W+1,W+2 NE tag of W Combination of Root word(R), POS Tag (P) and Suffix (S) (Bigram Features : R_P, P_S and R_S for current word) (7)	0.6863	0.9007
10	Root Word (R) of W, W-1 and W+1 POS Tag (P) of W,W-1,W-2,W+1,W+2 NE tag of W, Date Bigram Features : R_P, P_S and R_S for current word (6)	0.7118	0.9063
11	Root Word (R) of W, W-1 and W+1 POS Tag (P) of W,W-1,W-2,W+1,W+2 NE tag of W, Date Bigram Features : R_P, P_S and R_S for current word Person_suffix boolean feature, Location_Suffix boolean feature and Organization_Suffix boolean feature (13)	0.7461	0.9120

Table 3: CRF Features Set Accuracy

Following are the observations based on the CRF results shown in [Table 3].
1. Word window of [-2,1] works best without the first word feature and further increase in the window size reduced the overall F1-Score (Feature Set 1,2,3 and 4)
2. Results from Feature Set 6 shows that overall F1-Score was improved by 3% with the inclusion of NE tag information of the previous word.
3. Feature Set 8 shows that root and suffix of current, previous and next word were effective features in improving the F1 score by 6%.
4. Adding date boolean feature further improved the F1-score by 3%.
5. Finally, an overall F1-Score value of 74.61% was achieved by adding gazetteer lists which contained common suffix for Person, Location and Organization.

3.2 Experimental Results for Neural Networks

Following are the observations based on the results on test set.
1. Training word vectors was experimented using Skip gram model and Character enhanced encoding. Character enhanced encoding gave better clustering of words which was evaluated through randomly chosen 100 words through word relatedness computation by correlating results with human judgements.

2. We started training the NER model with 1 to 3 character window, then increasing the window to 5 improved our accuracy by 2%. Hyper parameters were tuned through the grid search.
3. The optimal results (minimum error loss rate) were obtained with 20 epochs.
4. Word vectors were initially trained using generic data due to lack of sufficient history specific data and later trained with history domain data which increased the accuracy of NER.
5. The accuracy of the neural networks without using any features and gazetteer list justifies that neural networks was able learn the language model.
6. The neural network training was time consuming due to which experimenting on parameters such as character window took more time. Currently, we have extracted 15 character level features and we can train the model to extract more features.
7. A pre-trained vector vocabulary of 134746 words was used by the neural networks. [Table 4] shows accuracy of prediction of known and unknown words with respect to word vector vocabulary.

Observation	Count	Accuracy(F1-micro)
Words Present & correct	11333	89.937%
Words Present & wrong	1268	10.003%
Words Absent & correct	3269	87.664%
Words Absent & wrong	460	12.322%

Table 4: Accuracy for known and unknown words

3.3 Comparison Results for CRF and NN

Named Entity	**CRF**			**NN**		
	Precision	**Recall**	**F1 score**	**Precision**	**Recall**	**F1 score**
PER	81.92	73.66	77.57	77.07	80.84	78.91
LOC	81.82	69.93	75.41	78.96	74.17	76.49
COM	79.13	68.77	73.58	77.98	65.19	71.01
DATE	87.67	75.79	81.30	80.73	67.18	73.33
NUM	80.43	74.50	77.35	78.21	79.48	78.84
ORG	79.80	48.77	60.54	69.7	50.27	58.41
ROL	67.06	47.61	55.68	60.06	47.84	53.26
NNE	93.57	97.44	95.47	94.84	96.56	95.69

NE Detection	75.69		79.23	
Overall NER Accuracy	Micro-F1	91.20	Micro-F1	91.11
	Macro-F1	74.61	Macro-F1	73.54

Table 5: Evaluation Results for Named Entity Tags

Following are the key observations for the results in [Table 5]

PER

Using CRF model, the F1-Score for person entities was improved when root and suffix features were added. It can be further improved by creating boolean feature that is set to 1 when the current word is in the predefined list of first name of the person names. NN model was able to predict person named entities accurately as Bi-LSTM automatically considered previous and next words as features.

LOC

Initially, CRF model did not recognize the newly encountered locations correctly with word features and named entity features. To overcome this issue, we included a list of words containing common location suffix(Nagar, Pura, Halli, Pradesh). The F1-Score improved after adding the boolean feature that is set to 1 for the current word if it contains the location suffix. NN model was able to give almost same accuracy as CRF without any feature specified. Morpheme features recognized by CNN was helpful in identifying location.

COM

CRF identified the community correctly by using the prefix of the word. For example, ಬ್ರಿಟಿಷರಿಂದ (by the British) and ಬ್ರಿಟಿಷರಿಗೆ (to the British) were present in the training data and CRF tagged ಬ್ರಿಟಿಷರೊಂದಿಗೆ (with the British) correctly. NN model was able to cluster community entities with more accuracy using the CNN character level features with additional train data.

DATE

The Date boolean feature which identifies whether the word contains year or date (month/day) expressed using Kannada numerals, improved the accuracy by 3%. Training the NN model trained with 100K words gave better accuracy of 86.2%, but with increased data, accuracy dropped to 67.17%.

NUM

CRF identified the number named entities correctly when specified as numeric digit format but, did not identify when specified as Kannada numeral (೧ -1). To counter this issue, we specified the boolean digit feature that is set to 1 if word contains Kannada numeral. This resulted in improving the accuracy for number entities by 6%. NN model trained with 100K words gave accuracy of 87.75%, but with increased data, accuracy dropped to 79.74%.

ORG

For both the models, optimal accuracy was not achieved due to fewer(1.5% of corpus) organization tags in the corpora and absence of pre-trained vectors for this entity.

ROL

The named entities with role tag was limited (2% of corpus). Also, in the confusion matrix, we observed most of the role data was clustered under not named entity for unknown words.

4 DISCUSSION

The results of CRF and Neural Networks for NER in Kannada history domain shows the impact of certain factors

1. Features such as Boolean Date feature, Previous NER tag and Root word have significant positive impact on the entity recognition.
2. The sparseness of certain named entities in the corpus such as ROL and ORG affects the final F1 score.
3. The inter-annotator accuracy, POS tagger and morphalizer determine the correctness of tagging of the named entities

4.1 Contribution

We are planning to publish the tagged NER corpus of 145K tokens with inter-annotator accuracy of 87% as contribution to Kannada NLP community. Also, the CRF model can be easily extended to other domains. The Neural Network pre-trained word vectors on 8M words is useful for building any neural network language model. The model from CNN is a by-product for morpheme analysis and can be used to overcome the existing ambiguities in Indic Library Morfessor.

4.2 Future Research

The motive of the project was to develop a reliable NER system for history domain in Kannada and conduct the experiments on both CRF and Neural Networks. The learning curve was deep and we have successfully executed the experiments for both CRF and Neural Network techniques. The base model can be extended to other domains in Kannada apart from history. We are planning to improve the CRF model by supplying more data and experiment with additional features. Currently, we have used one hot encoding as an input to CNN model and we want to experiment with pre-trained character vectors as input and evaluate the accuracy of resulting Neural Network model. The models that we have developed can be used to build useful applications such as Question Answering System and Speech recognizers.

5 REFERENCES

[1] S Amarappa and Dr. S V Sathyanarayana. A Hybrid approach for Named Entity Recognition, Classification and Extraction (NERCE) in Kannada Documents. Association of Computer Electronics and Electrical Engineers, 2013

[2] R Vijayakrishna and L Sobha. Domain Focused Named Entity Recognizer for Tamil Using Conditional Random Fields. Asian Federation of Natural Language Processing, 2008

[3] S Amarappa and Dr. S V Sathyanarayana. KANNADA NAMED ENTITY RECOGNITION AND CLASSIFICATION (NERC) BASED ON MULTINOMIAL NAÏVE BAYES (MNB) CLASSIFIER

[4] S Amarappa and Dr. S V Sathyanarayana. Kannada Named Entity Recognition and classification using Conditional Random Fields. IEEE, 2015

[5] Bhat Suma. Morpheme Segmentation for Kannada Standing on the Shoulder of Giants. Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing, 2012.
[6] Pallavi and Dr. Anitha S Pillai. Named Entity Recognition for Indian Languages: A Survey. International Journal of Advanced Research in Computer Science and Software Engineering, 2013.
[7] Asif Ekbal, Rejwanul Haque and Sivaji Bandyopadhyay. Named Entity Recognition in Bengali: A Conditional Random Field Approach.
[8] Xinxiong Chen1, Lei Xu1, Zhiyuan Liu1, Maosong Sun, Huanbo Luan. Joint Learning of Character and Word Embeddings. National Lab for Information Science and Technology, Tsinghua University, Beijing, China.
[9] Rudra Murthy V, Mitesh Khapra and Dr. Pushpak Bhattacharyya. Sharing Network Parameters for Crosslingual Named Entity Recognition. Indian Institute of Bombay.
[10] Minh-Thang Luong, Richard Socher and Christopher D. Manning. Better Word Representations with Recursive Neural Networks for Morphology.
[11] Daniele Bonadiman, Aliaksei Severyn and Alessandro Moschitti. Deep Neural Networks for Named Entity Recognition in Italian.
[12] Siwei Lai, Kang Liu, Liheng Xu and Jun Zhao. How to Generate a Good Word Embedding? National Laboratory of Pattern Recognition (NLPR) Institute of Automation, Chinese Academy of Sciences, China.
[13] Corpus References : The EMILLE/CIIL Corpus (ELRA-W0037). A joint venture The EMILLE project (Enabling Minority Language Engineering), Lancaster University, UK, and the Central Institute of Indian Languages (CIIL), Mysore, India.
[14] Corpus References : High School Textbooks on Indian History.
[15] Tool Reference : Kannada POS Tagger by Siva Reddy-
https://github.com/anoopkunchukuttan/indic_nlp_library
[16] Tool Reference : Indic NLP Library by Anoop Kunchukuttan-
https://github.com/anoopkunchukuttan/indic_nlp_library
[17] Tool Reference : CRF ++ toolkit - https://taku910.github.io/crfpp/

5 DIVISION OF TASK

Task	**Team Members**	**Start Date**	**End Date**
Corpus Generation and Tagging	Harshitha, Nagarjun	02/26/2017	03/06/2017
Method Analysis and Design	Sruthi, Harshitha, Komal, Nagarjun	03/07/2017	03/15/2017
NER Model Training	Komal, Sruthi	03/15/2017	03/20/2017
Implementation	Sruthi, Harshitha,	03/20/2017	03/31/2017

	Komal, Nagarjun		
Evaluation	Komal, Harshitha	04/01/2017	04/05/2017
Final Report	Sruthi, Harshitha, Komal, Nagarjun	04/06/2017	04/15/2017

Github Repository - https://github.com/smravi/NER-Kannada
Word count of the document - 4000