Khushleen@IECSIL-FIRE-2018: Indic Language Named Entity Recognition Using Bidirectional LSTMs with Subword Information

Khushleen Kaur

Shaheed Udham Singh College of Engineering and Technology
Kharar Banur Highway, Tangori, Punjab, 140306, India
khushleendhanoa@gmail.com

Abstract. Named Entity Recognition generally requires large amount of tagged corpus to build a high performing system. The representation has always been a bottleneck in NERs success. The NER subtask by IECSIL had enough data for algorithms to learn semantic representation as well as apply deep learning models. The current work uses a subword aware word representation for generating representations. These embeddings are further used with a bidirectional LSTM for building an NER system. The system performed well for all the Indian languages and stood among top three submissions.

Keywords: Indic Languages · Bidirectional LSTMs · Subword Information · word embeddings

1 Introduction

The most celebrated approaches for Named Entity Recognition has either been Conditional Random Fields or Support Vector Machines with feature engineering [1]. The recent advancements in representation learning as well as neural network algorithms have opened doors for various new possibilities. A word representation learned on sufficient data followed by a suitable deep learning algorithm can outperform the existing state of the art approaches.

The representations learning algorithms plays a crucial role in determining system performance. The word embedding methods like word2vec by Mikolov et. al [2] and GloVe by Pennington et. al [3] has helped achieve much better results than ever before. Both of these famous embedding algorithms doesn't take into account the subword information. The vector representation proposed by [4] is an extension to Mikolovs skip gram model but it includes character n-grams which subsequently represent words as sum of these character n-grams. These character level methods also makes it possible to learn embeddings for rare words which are not generally poorly trained. The representation techniques alone cannot win the battle of better performance for us. It requires a suitable algorithm which can leverage afaorementioned character level subword information. The sequence-in sequence-out deep learning architecture of Recurrent Neural Networks (RNNs), more specifically, Long Short Term Memory (LSTM) is just the right choice for such requirement. We used the word embeddings with subword information followed by Bidirectional LSTM for our architecture development.

Since the shared task [9] focused only on Indian languages mainly Hindi, Tamil, Kannada, Malayalam & Telugu, considering subword information helps learn the morphological word representations.

2 Corpus Statistsic

The corpora provided by the shared task organizers was in 5 languages, namely, Hindi, Tamil, Malayalam, Kannada & Telugu [8]. The corpus size was sufficient to leverage deep learning techniques. The corpus per language was segregated into three parts such as Training which was 60%, Testing phase-1 20 % and Testing phase-2 20 %. The phase-2 test corpus was used to finally rank the submitted systems. The training & testing statistic are provided in the Table 1.

S. no.	**Language**	**Train (# of words)**	**Test-1 (# of words)**	**Test-2 (# of words)**
1	Hindi	1548570	519115	517876
2	Tamil	1626260	542225	544183
3	Malayalam	903521	301860	302232
4	Kannada	318356	107325	107010
5	Telugu	840908	280533	279443

Table 1. Corpus statistic per language

3 Methodology

3.1 Word embeddings with subword information

The neural network based word representations were proposed by Collobert and Weston [5] which used a simple feed forward network. It doesn't really captures a long range relationships among words. The distributional representation technique proposed by mikolov more recently uses a log bilinear model to learn the continous word representations. It only works when you have a very large data to learn the representations efficiently.

The aforementioned techniques represent each word in the vocabulary as a unique vector. It doesn't allow parameter sharing among the words. The morphological structure is hard to capture this way since agglutinative languages contains many word forms that hardly occur in the training data. A good representation can be learned if all these word forms are considered while learning continous vector representation. Since it is not possible to have all the word forms for morphologically rich languages in training corpora, using character level information will help impove the word representation. It is observed by [4] that including characeter level information does help include rare words or out of vocabulary word representations from the given corpora. Basically, a word is represented by vector sum of its character n-grams. The scoring function thus obtained is,

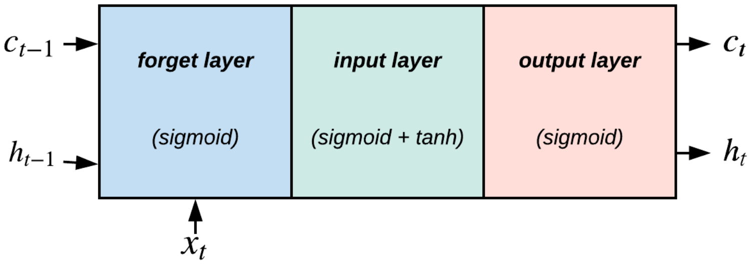
$$s(w,c) = \sum_{g \epsilon G_w} z_g^t v_c \tag{1}$$

where, w si a word which is represented as a bag of character n-grams, G refers to n-gram collection and z_g is vector representation for each n-gram g.

3.2 Bidirectional Long Short Term Memory (BiLSTM)

The Recurrent Neural Network [6] family of neural networks is the de-facto standard to use when you have a sequence to deal with. Unlike Convolutional Neural Network, they take dynamic size sequence and also retain long range dependencies. The vanilla RNN suffers from the vanishing/exploding gradient drawbacks which maked it hard for these algorithms to learn longer-range dependencies.
The LSTM [7] networks were designed for a situation like this. It is a particular type of RNN which works better than RNNs owing to its more powerful update equation and a slightly different backpropagation. LSTM has the capability of selectively remember (add) or forget (remove) information owning this feature to carefully regulated gates, namely, input gate, forget gate & output gate.One additional feature in LSTMs is the another amazing ability of reading the input sequence either unidirectionally or bidirectionally. In bidirectional case, it reads the sequence left to right as well as right to left. It does takes more memory but it has proven better results. The basic structure of a LSTM cell is depicted in Fig 1.

Fig. 1. Basic LSTM cell structure



3.3 Subword word embedding with Bidirectional LSTM

The implementation for this work was completed in two steps. First the text corpora for all the languages was processed throught the word embedding module fastText[1] per languages. The parameters used for all the languages were kept same in order to make a unified model. These 300 dimensional continous vector representations were then fed to a 2 layer BiLSTM with each layer having 64 neurons each. The number of epoch asn batch size used were 35 and 128 respectively. The BiLSTM architecture topology for all the languages were kept same to make the model unified and language independent.

4 Results

The unified system developed for the shared task performed well across all the languages. The results reported by the organizers are shown in Table 2. It can be observed from the results (in bold) that the system performance was comparative for all the languages. It means the same model can be ported to other Indian languages.

Team	Runs	Malayalam	Kannada	Hindi	Tamil	Telugu	Average
hilt	2	92.1	93.17	94.35	91.79	92.47	92.776
raiden11	1	89.6	92.33	91.19	87.26	89.19	89.914
SSN_NLP	3	95.05	94.21	95.95	94.66	95.4	95.054
hilt	2	92.12	93.17	94.28	91.79	92.47	92.766
am905771	2	88.89	89.85	94.47	90.4	90.04	90.73
idrbt-team-a	1	96.58	96.79	97.82	96.18	97.68	97.01
SSN_NLP	2	95.28	95.76	96.51	94.9	96.81	95.852
khushleen	**1**	**96.18**	**96.45**	**96.85**	**95.83**	**96.78**	**96.418**
Ajees	1	96.86	97.09	97.65	96.85	97.69	97.228
hariharanv	1	95.63	95.79	96.67	0	96.39	76.896
rohitkodali	1	0	96.85	98.06	0	97.53	58.488
am905771	3	89.13	89.88	94.92	90.47	90.32	90.944
SSN_NLP	1	95.28	95.8	96.68	94.91	96.81	95.896
am905771	1	89.04	89.53	94.45	90.46	90.04	90.704

Table 2. Official shared task results

5 Conclusion

The NER problem from NLP requires a lot of tagged corpus to build a decent system. In this shared task, used a subword (character n-grams) aware word representation for generating representations. These embeddings were further

[1] https://fasttext.cc/

used with a bidirectional LSTM with common setting acrosss all the languages to build a unified model. The developed system performed well and gave consistent results across all the Indian languages. The same model can be used for other Indian languages.

References

1. Ratinov, Lev and Roth, Dan: Design challenges and misconceptions in named entity recognition. Proceedings of the Thirteenth Conference on Computational Natural Language Learning, 147–155 (2009)
2. Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey: Efficient estimation of word representations in vector space. In: arXiv preprint arXiv:1301.3781 (2013)
3. Pennington, Jeffrey and Socher, Richard and Manning, Christopher: Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 532–1543 (2014)
4. Bojanowski, Piotr and Grave, Edouard and Joulin, Armand and Mikolov, Tomas: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
5. Collobert, Ronan and Weston, Jason: A unified architecture for natural language processing: Deep neural networks with multitask learning. Proceedings of the 25th international conference on Machine learning 160–167 (2008)
6. Hochreiter, Sepp and Schmidhuber, Jürgen: Long short-term memory. Neural computation:vol 9 1735–1780 (1997)
7. Lipton, Zachary C and Berkowitz, John and Elkan, Charles: A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019 (2015)
8. Barathi Ganesh, H B and Soman, K P and Reshma, U and Mandar, Kale and Prachi, Mankame and Gouri, Kulkarni and Anitha, Kale and Anand Kumar, M: Information Extraction for Conversational Systems in Indian Languages - Arnekt IECSIL. Forum for Information Retrieval Evaluation (2018)
9. Barathi Ganesh, H B and Soman, K P and Reshma, U and Mandar, Kale and Prachi, Mankame and Gouri, Kulkarni and Anitha, Kale and Anand Kumar, M, Kale: Overview of Arnekt IECSIL at FIRE-2018 Track on Information Extraction for Conversational Systems in Indian Languages. FIRE (Working Notes) (2018)