Named Entity Recognition for Code-Mixed Indian Corpus using Meta Embedding

Ruba Priyadharshini
Department of Mathematics
Saraswathi Narayanan College
Madurai, India
rubapriyadharshini.a@gmail.com

Bharathi Raja Chakravarthi*
Data Science Institute
National University of Ireland
Galway, Ireland
bharathi.raja@insight-centre.org

Mani Vegupatti*
Data Science Institute
National University of Ireland
Galway, Ireland
mani.vegupatti@insight-centre.org

John P. McCrae*
Data Science Institute
National University of Ireland
Galway, Ireland
John.MCrae@insight-centre.org

***Abstract*—In this paper, we utilize the pre-trained embedding, sub-word embedding and closely related languages of languages in the code mixed corpus to create a meta-embedding. We then use the Transformer to encode the code mixed sentence and use Conditional Random Field to predict the Named Entities in the code-mixed text. In contrast to classical Named Entity recognition where the text is monolingual, our approach can predict the Named Entities in code-mixed corpus written both in the native script as well as Roman script. Our method is a novel method to combine the embeddings of closely related languages to identify Named Entity from Code-Mixed Indian text written using native script and Roman script in social media.**

***Index Terms*—code-mixing, code-switching, Indian code mixing, embedding, meta embedding, named entity recognition, conditional random field**

I. Introduction

Named Entity Recognition (NER) helps to locate the single word or multi-word nouns that could uniquely represent an entity, which belongs to a specific category with distinct representation but might have different mentions in the text. NER is one of the most essential tasks in multiple applications that include Natural Language Processing (NLP), Information Extraction (IR) and Machine Translation, to name a few. For carrying out NER task with desired results, large amount annotated data is mandatory, however for code mixed data, the availability of annotated data is limited or nil. Hence generalization using word embedding and meta-embedding the multiple languages will be beneficial to tackle this problem.

Code mixing is a natural means of communication in the Indian urban societies or multilingual communities across the world [1]. This is a concept where the lexical units and grammar of more than one language is used interchangeably as if they are monolingual syntactic elements. Code mixing frequently happens between languages possessing different scripts, for reusability of the research work, in this work we use Romanized scripts for Non-Roman script language also to analyze the text.

Word Embedding is used to project the learned word at a high dimensional vector space for generalization of the meaning [2]. The words having related meaning are projected closely in the same vector space using Word Embedding and enables vector computation to derive relations [3]. The advantage of Word Embedding is a number of dimensions required to model them are much lesser than the original vocabulary and they capture syntactic and semantic details in a simpler way [4]. These characteristics motivate the use of the pre-trained word embedding for this work.

Closely related languages are the group of languages, those share similarities at the lexical, syntactic and structural level. In closely related languages, often it is possible for speakers to carry out interaction with ease using their respective languages without the need for switching to another language [5]. The quantum of similarity will vary between the languages based on the level of relatedness and mostly they share the roots for lexical units and shared vocabulary [5], [6]. This allows code-mix in these languages more naturally. We will also exploit the relatedness to combine the embedding of this group of languages for NER task.

In NLP, building or choosing the suitable Word Embedding for the current task is a primary step. Meta-Embedding is a technique where different embedding is combined together to make use of best out of each them and the augmentation of capabilities leads to better results than left individual [7]. In the advent of the availability of pre-trained embedding, code-mix in the text and close-relatedness between languages, meta-embedding of pre-trained word embedding provides better results than monolingual word embedding [8]. This technique can be viewed as an ensemble, while the most important aspect is neural networks are used to build this model by taking individual embedding as input and the learned representation is projected as output [9].

Byte Pair Encoding (BPE) can be used as an efficient

*Authors are supported by a research grant from Science Foundation Ireland, co-funded by the European Regional Development Fund, for the Insight Centre under Grant Number SFI/12/RC/2289.

978-1-7281-5197-7/20/$31.00 ©2020 IEEE

method for reducing the representation length in the word segmentation, which is found beneficial to find sub-word information to take advantage of resources from closely related languages. Studies showed that the model's performance is improved when BPE is used in the training of neural networks for Machine Translation and other NLP tasks [10]. While performing the word representation for Closely related languages, the use of BPE improves the level of encoding. The reason is BPE compresses the word using most commonly used character sequences and closely related languages share the common root for the lexical units [11].

We propose NER for Code-Mixed Indian languages, which learns how to combine different pre-trained monolingual embedding in word and sub-word level into a single language lexical representation without word-level language tags of the corpus. To develop the hierarchical meta-embedding we use Transformer [12] for encoding the data combined with Conditional Random Field (CRF) for predicting the Named Entity tags. Our experiment results show better results compared to the baseline on monolingual embeddings.

II. Related Work

Named Entity Recognition for monolingual corpus has achieved much attention from natural language processing experts and produced state-of-results from CoNLL-2003 NER shared task [13], systems based on hand featured engineering machine learning approaches [14]–[17] to recent deep learning-based approaches [18]–[21]. However, all these approaches only consider the monolingual corpus and were costly for new languages and domains. Our work studies the Named Entity Recognition in code mixed corpus.

Several researchers have investigated the code-mixing from the linguistic motivations [22] expressed that while communications multilingual speakers often blend their languages frequently. Recently methods have been developed to automatically identify the language of code-mixed text at word level [23], [24] and Part of Speech tagging at word level [25]–[27] but building corpus for the language identification for new languages was expensive. If the code mixed text of the corpus was not written in native scripts that make even more challenging. Previous work addresses the code-mixing phenomenon for recognising the language and word form to select the suitable tagging or classification tool.

Previous works have extensively studied the word representation [28], [29], sub-word level [30]–[33] and showed that pre-trained word embedding improves the text classification results [34]. Creating meta embedding from pre-trained embedding has received much attention recently. Several methods proposed to create meta-embedding monolingual embedding [9], [35]. Our proposed method is similar to the recent work on Named Entity recognition for English-Spanish [36], [37] but their method does not work for the languages which use the native script as well as Roman scripts. They only considered the European languages.

III. Methodology

Commonly, Named Entity Recognition systems to use a single type of word embeddings. We propose a method to combine word, sub-word form embeddings written in the native script as well as written in Roman (transliteration) script to create a meta-embeddings.

A. Embedding

Let us consider n number of pre-trained word embeddings, denoted by E_1,....E_n. We denote the dimensionality of E_k embedding by d_k. The set of words in vocabulary is denoted by V_1,...V_n. Let the set union of V_1,...V_n be $W = w_1, ...w_n$ containing n words. Each word can be tokenized into a list of subwords $S = [s_1, ..., s_n]$. We generate a meta-representation by taking vector representation of multiple pre-trained monolingual embeddings in words and sub-words. We do an experiment in four settings as we follow the paper [36], [37]. First, we concatenate the word embeddings of all the languages by merging the dimension of each language word representation. This is not an efficient way of doing meta-embeddings. Second, weights of each embedding E_k is calculated using attention mechanism, where each embedding E_k is vector of $d-$dimension constructed by a nonlinear function by projection on a layer that is fully connected.

B. Named Entity Recognition

NER can be seen as a problem at word-level, given a corpus of tokens and tags associated with the tokens, the program has to train on the corpus and predict the tags given the tokens. A NE may span to multiple words since (B-) and (I-) tags specify the start and inner of the NE for position indication. (PER, ORG, LOC) tags are used to specify the Person, Organisation and Location correspondingly NE type tags. In addition, an O tag indicates it is not a Named Entity.

The transformer has recently achieved superior performance in NLP using the CRF, the undirected graphical model that for the chosen output node's value based on the input node's value computes the conditional probability. It calculates the dependencies across tag labels such (B-) tag and (I-) tag and learns correlations between the current label and previous labels. It calculates $P(S|O)$ where S = $< s_1, s_2, ..., s_T >$ is a set containing sequence of states and O = $< o_1, o_2, ..., o_T >$ is a set containing sequence of observations [38] as

$$P_{\wedge}(s|o) = \frac{1}{Z_0} \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k \times f_k\left(s_{t-1}, s_t, o, t\right)\right) \quad (1)$$

Training facilitates the learning of weight λ_k from feature $f_k\left(s_{t-1}, s_t, o, t\right)$. Normalization factor is calculated to ensure the constraint that sum of conditional probabilities are always at max one.

$$Z_0 = \sum_{s} \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k \times f_k\left(s_{t-1}, s_t, o, t\right)\right) \quad (2)$$

978-1-7281-5197-7/20/$31.00 ©2020 IEEE

To train the CRF, penalized log-likelihood is the objective function to be maximized for the selected S (state sequence) and O (observation sequence)

$$L_{\wedge} = \sum_{i=1}^{N} \log \left(P_{\wedge} \left(s^{(i)} | o^{(i)} \right) \right) - \sum_{k} \frac{\lambda_k^2}{2\sigma^2} \tag{3}$$

To predict the named entities we use Transformer-CRF a transformer-based encoder and Conditional Random Forest predicter. We take the meta-embedding use this in the Transformer to encode the sentence and CRF takes the encoded information from the Transformer to predict the tags of Named Entity.

IV. Experiment

We use FastText word embedding trained from Common Crawl and Wikipedia [30] for English and Hindi-Devanagari script (native script for Hindi). We also add the English Twitter GloVe word embeddings since the NER data is from Twitter. To create a word embedding for Hindi, Bengali, Marathi, and Panjabi in the Roman script we downloaded the latest Wikipedia dump from Wiki-dump website [1]. The dump files are extracted to get only text part of the website using Wikipedia Extractor [2]. Then the text file is transliterated using indic-trans library [3]. The transliterated corpora are then trained using fastText to create a pre-trained word embeddings. We used skip-gram model to produce vectors, where dimensions are 300 and parameters are default.

We used Hindi-English code mixed texts from the Twitter corpus released by Singh et. al [39]. The corpus was created using select domains namely politics, sports and social belongs to a subcontinent (India) for a duration of 8 years. Mining of these tweets was done by using specific hash-tags. The corpus contains 3,638 code-mixed tweets and annotated using CONLL standard tags (Loc) Location, (Org) Organisation, (Per) Person, (O) Other to tag while in annotation stage. Tweets are pre-processed and annotated as per the 6 Named Entity tags and 7th Other tag described in the Table I. This data is split into 3000, 338 and 300 entries for the train, development, and test set respectively. A model trained with parameters, Optimizer set as Noam, multilingual dropout setting set to 0.1, and transformer constructed using block of four layers, 200 as hidden size, and number of heads set to 4. The starting value of the learning rate is 0.1.

TABLE I
NER-Tags for the dataset used

Tag	Explanation
B-Per	Indicates the Beginning of a Person's name.
I-Per	Indicates the intermediate of a Person's name.
B-Org	Indicates the Beginning of a Organization's name.
I-Org	Indicates the intermediate of a Organization's name.
B-Loc	Indicates the Beginning of a Location's name.
I-Loc	Indicates the intermediate of a Location's name.
Other	Indicates all the word not falling in any of the above 6.

V. Results and Discussion

Precision, Recall, and F-Score are the scores used for the evaluation of the Named Entity Recognizer. Precision is the ratio between extracted items that are relevant with reference to the total items extracted, whereas Recall is the ratio between extracted items that are relevant with reference to the total items that are relevant in the given text [40], [41]. For calculation, we used Precision as the ratio of TP over the sum of TP and FP, and recall as the ratio of TP over the sum of TP and FN. To compare the results using a single measure as a performance indicator, we will use F-Score calculated as harmonic mean using Precision and Recall [42].

$$\text{Precision } = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\textbf{F-Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

where, TP is True Positive, FP is False Positive, and FN is False Negative.

TABLE II
F-Score for Named Entity Recognition using hierarchical meta-emdingings for English-Hindi code-mixed corpus. E

Model	En+Hi	+Pa+Mr+Bn
Concat	2.90	2.85
Linear	2.18	2.16
Meta-Emb-Word	2.24	2.18
Meta-Emb-Word-BPE	47.51	49.79

Table II show the result for the English-Hindi Code-Mixed dataset. From the Table II, it is clear that adding pre-trained embedding along with the BPE embedding outperforms the baseline. Our baseline results were very poor compared with state of art results of NER. However, our approach does not require word-level annotation or hand-crafted feature selection to perform the task of Named Entity Recognition. We observed that the word vectors from FastText alone is not enough to improve the results of the task.

There is an increase in F1-score between English-Hindi and English-Hindi+Punjabi+Marathi+Bengali. This improvement is not possible in the baseline or word only model because the closely related language information cannot be captured by the word only model. The BPE takes advantage from closely related languages from the fact that they share cognates.

VI. Conclusion

In this paper, proposal is made to use NER in code-mixed Indian languages to combine pre-trianed emebedding. We analyzed the closely related language hierarchical meta-embedding for Named Entity Recognition. We did it for English-Hindi code mixed corpus. The Hindi was written in

[1] https://dumps.wikimedia.org/
[2] https://github.com/attardi/wikiextractor
[3] https://github.com/libindic/indic-trans

978-1-7281-5197-7/20/$31.00 ©2020 IEEE

Roman Script(native script is Devenagri) so we transliterated the corpus and did the experiments. We found that hierarchical meta-embedding with sub-word information (BPE) produced a better F1-Score. We also found that adding embedding of closely related languages also improves the F1-Score.

References

[1] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. Bali, "Language modeling for code-mixing: The role of linguistic theory based synthetic data," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1543–1553.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[4] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2013, pp. 746–751.

[5] C. Gooskens, V. J. van Heuven, J. Golubović, A. Schüppert, F. Swarte, and S. Voigt, "Mutual intelligibility between closely related languages in europe," *International Journal of Multilingualism*, vol. 15, no. 2, pp. 169–193, 2018.

[6] Y. Scherrer and B. Sagot, "Lexicon induction and part-of-speech tagging of non-resourced languages without any bilingual resources," in *Proceedings of the Workshop on Adaptation of Language Resources and Tools for Closely Related Languages and Language Variants*, Sep. 2013.

[7] W. Yin and H. Schütze, "Learning word meta-embeddings," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug. 2016, pp. 1351–1360.

[8] D. Kiela, C. Wang, and K. Cho, "Dynamic meta-embeddings for improved sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Oct.-Nov. 2018, pp. 1466–1477.

[9] J. Coates and D. Bollegala, "Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Jun. 2018, pp. 194–198.

[10] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug. 2016, pp. 1715–1725.

[11] A. Kunchukuttan and P. Bhattacharyya, "Learning variable length units for SMT between related languages via byte pair encoding," in *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, Sep. 2017, pp. 14–24.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.

[13] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, 2003, pp. 142–147.

[14] J. R. Finkel and C. D. Manning, "Joint parsing and named entity recognition," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Jun. 2009, pp. 326–334.

[15] G. Zhou and J. Su, "Named entity recognition using an hmm-based chunk tagger," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02, 2002, pp. 473–480.

[16] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, "Named entity recognition through classifier combination," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, ser. CONLL '03, 2003, pp. 168–171.

[17] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticæ Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Nov. 2011.

[19] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[20] R. Panchendrarajan and A. Amaresan, "Bidirectional LSTM-CRF for named entity recognition," in *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, 1–3 Dec. 2018.

[21] B. Johansen, "Named-entity recognition for Norwegian," in *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 30 Sep. – 2 Oct. 2019, pp. 222–231.

[22] E. E. Papalexakis, D.-P. Nguyen, and A. Doğruöz, "Predicting code-switching in multilingual communication for immigrant communities," in *Proceedings of the First Workshop on Computational Approaches to Code Switching*, 10 2014, pp. 42–50.

[23] Z. Yirmibeşoğlu and G. Eryiğit, "Detecting code-switching between Turkish-English language pair," in *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, Nov. 2018, pp. 110–115.

[24] D. Mave, S. Maharjan, and T. Solorio, "Language identification and analysis of code-switched social media text," in *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, Jul. 2018, pp. 51–61.

[25] K. Ball and D. Garrette, "Part-of-speech tagging for code-switched, transliterated texts without explicit language identification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Oct.-Nov. 2018, pp. 3084–3089.

[26] V. Soto and J. Hirschberg, "Joint part-of-speech and language id tagging for code-switched data," in *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, 2018, pp. 1–10.

[27] V. Soto, N. Cestero, and J. Hirschberg, "The role of cognate words, pos tags and entrainment in code-switching." in *Interspeech*, 2018, pp. 1938–1942.

[28] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[29] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct. 2014, pp. 1532–1543.

[30] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[31] D. Torregrosa, N. Pasricha, M. Masoud, B. R. Chakravarthi, J. Alonso, N. Casas, and M. Arcan, "Leveraging rule-based machine translation knowledge for under-resourced neural machine translation models," in *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, 19–23 Aug. 2019, pp. 125–133.

[32] B. R. Chakravarthi, R. Priyadharshini, B. Stearns, A. Jayapal, S. S, M. Arcan, M. Zarrouk, and J. P. McCrae, "Multilingual multimodal machine translation for Dravidian languages utilizing phonetic transcription," in *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, 20 Aug. 2019, pp. 56–63.

[33] B. R. Chakravarthi, M. Arcan, and J. P. McCrae, "WordNet gloss translation for under-resourced languages using multilingual neural machine translation," in *Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation*, 19 Aug. 2019, pp. 1–7.

[34] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Apr. 2017, pp. 427–431.

[35] D. Bollegala, K. Hayashi, and K. Kawarabayashi, "Think globally, embed locally - locally linear meta-embedding of words," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2018, pp. 3970–3976.

[36] G. I. Winata, Z. Lin, and P. Fung, "Learning multilingual meta-embeddings for code-switching named entity recognition," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, Aug. 2019, pp. 181–186.

[37] G. I. Winata, Z. Lin, J. Shin, Z. Liu, and P. Fung, “Hierarchical meta-embeddings for code-switching named entity recognition,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp. 3539–3545.

[38] A. Ekbal, R. Haque, and S. Bandyopadhyay, “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.

[39] V. Singh, D. Vijay, S. S. Akhtar, and M. Shrivastava, “Named entity recognition for Hindi-English code-mixed social media text,” in *Proceedings of the Seventh Named Entities Workshop*, Jul. 2018, pp. 27–35.

[40] E. F. Tjong Kim Sang and F. De Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition,” in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147.

[41] A. Esuli and F. Sebastiani, “Evaluating information extraction,” in *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 2010, pp. 100–111.

[42] H. Schütze, C. D. Manning, and P. Raghavan, “Introduction to information retrieval,” in *Proceedings of the international communication of association for computing machinery conference*, 2008, p. 260.

978-1-7281-5197-7/20/$31.00 ©2020 IEEE