A Pre-trained Transformer and CNN model with Joint Language ID and Part-of-Speech Tagging for Code-Mixed Social-Media Text

by

Suman Dowlaqar, Radhika Mamidi

in

RANLP

Report No: IIIT/TR/2021/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2021

A Pre-trained Transformer and CNN model with Joint Language ID and Part-of-Speech Tagging for Code-Mixed Social-Media Text

Suman Dowlagar
LTRC
IIIT-Hyderabad
suman.dowlagar
@research.iiit.ac.in

Radhika Mamidi
LTRC
IIIT-Hyderabad
radhika.mamidi
@iiit.ac.in

Abstract

Code-mixing (CM) is a frequently observed phenomenon that uses multiple languages in an utterance or sentence. There are no strict grammatical constraints observed in code-mixing, and it consists of non-standard variations of spelling. The linguistic complexity resulting from the above factors made the computational analysis of the code-mixed language a challenging task. Language identification (LI) and part of speech (POS) tagging are the fundamental steps that help analyze the structure of the code-mixed text. Often, the LI and POS tagging tasks are interdependent in the code-mixing scenario. We project the problem of dealing with multilingualism and grammatical structure while analyzing the code-mixed sentence as a joint learning task. In this paper, we jointly train and optimize language detection and part of speech tagging models in the code-mixed scenario. We used a Transformer with convolutional neural network architecture. We train a joint learning method by combining POS tagging and LI models on code-mixed social media text obtained from the ICON shared task.

1 Introduction

In bilingual and multilingual communities, code-mixing or code-switching occurs when a person alternates languages below the phrase level inside a sentence or an utterance. Code-mixing(CM) is generally observed in informal settings such as casual conversations or social media text.

Code-mixing is defined as mixing phrases, words, and morphemes of one language into another language (Myers-Scotton, 1997).

Language identification (LI) and part of speech (POS) tagging are the fundamental steps in processing any code-mixed sentence. LI deals with resolving the language ambiguity of each word in a code-mixed text. POS tagging involves assigning a part of the speech label for each word in a sentence based on its syntactic and semantic information. It helps analyze the grammatical structure of the sentence. Both Language Identification and POS tagging are sequence labeling tasks as they tag each word in a sentence by its corresponding language and POS tags, respectively.

While processing a monolingual text, the primary step to understand the sentence's grammatical structure would be POS tagging. However, in the code-mixed scenario, we must consider multilingual phenomena, i.e., each word's language while POS tagging. Similarly, POS tagging helps capture a better grammatical structure of the text. Moreover, identifying the grammatical structure can improve language identification. In the code-mixing scenario, two tasks go hand in hand. Thus, a joint learning model on POS tagging and LI will considerably enhance the code-mixed text analysis.

Recently the transformer models with transfer learning such as BERT (Devlin et al., 2018) achieved state-of-the-art accuracy in sequence classification tasks. An evaluation benchmark on code-mixed datasets - GLUECoS (Khanuja et al., 2020) stated that a modified version of BERT (Devlin et al., 2018) called mod-mBERT that was fine-tuned on synthetically generated code-switched data outperformed on all the code-mixed datasets.

Recently, BERT with ensemble models has shown improved performance on text classification tasks (Dowlagar and Mamidi, 2021; Safaya et al., 2020). To improve our tasks' performance, we have used a convolutional model. The convolutional approach learns a compositional structure in the sequences more efficiently since the representations are built on hierarchy.

In the code-mixed social media text, the native words are often written in the Roman script. The introduction of such non-standard transliterations will add complications while processing the

text. It is necessary to deal with these complications as they might introduce errors in further processing steps. To tackle such problems, we back-transliterated the Roman words into their native script.

This paper presents a pre-trained transformer encoder with convolutional neural network architecture for POS tagging and language identification of Code-Mixed Social-Media text. The model uses sub-word level input representation to handle morphologically rich words.

Our contributions are as follows:

1. We pre-process the data to deal with variations in spelling and transliterations.

2. We propose a transfer learning-based approach to jointly model LI and POS tagging tasks, achieving state-of-the-art accuracy on the ICON 2016 shared task dataset.

3. We design a BERT with convolutional neural network architecture for LI and POS tagging tasks. Our analyses confirm it is a better alternative than the joint Bi-LSTM model.

The paper is organized as follows. Section 2 presents a survey of related works on language identification and POS tagging in a code-mixed social media text. Section 3 introduces the proposed approach of jointly modeling the language identification and the part-of-speech tagging task for code-mixed social media text. In Section 4, we present the experimental setup and performance of our joint model. Section 5 concludes the work.

2 Related Work

Since the last decade, LI and POS tagging for code-mixed text has been a topic of interest in the field of natural language processing (NLP).

CM language identification :(Aguilar and Solorio, 2019) used a large pre-trained model ELMo, and adapted it to code-switching settings to obtain contextually rich embeddings. The paper used a Bi-LSTM CRF for language identification. (Mave et al., 2018) compared different word-level language identification systems for code-switched Hindi-English data and a standard Spanish-English dataset and found that the CRF model works the best on the given datasets. (Gundapu and Mamidi, 2018) performed LI using CRF on Telugu-English CM data. (Barman et al., 2014; Solorio et al., 2014) used SVM and CRF for language identification on CM data.(Rijhwani et al., 2017) used HMM for language identification on seven languages.

CM POS tagging: (Bhattu et al., 2020a) addressed the problem of prediction of POS tags for OOV words in low resource languages using character-based word embedding as input features to a Bi-LSTM and CRF network. (Ball and Garrette, 2018) used a meta embedding approach for the part of speech tagging where the word is represented in both code-mixed languages. Thus, it maintains embeddings for each language and is processed appropriately at inference time without committing to one or the other language. It used a Bi-LSTM model for POS tagging. (Jamatia et al., 2015) presented POS tagging results on English-Hindi social media text using various ML and deterministic approaches, among which CRF gave the best results.

Joint Learning: The joint learning models have been studied for the code-switched LI and POS scenarios. (Soto and Hirschberg, 2018) proposed a joint learning approach for POS tagging and LI using recurrent neural networks. (Barman et al., 2016) used a factorial CRF for joint modeling of POS tagging and language identification.

To the best of our knowledge, a joint learning model with BERT and CNN architecture for language identification and POS tagging on code-mixed data is not yet analyzed.

3 Proposed Model

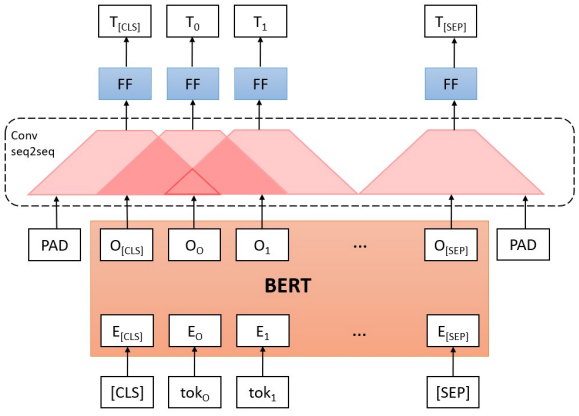

Figure 1: BERT with convolutional model.

In this section, we briefly describe the BERT and convolutional models. We then introduce the proposed joint model for POS tagging and LI of CM social media text. The architecture of the proposed model is given in figure 1.

3.1 Pre-processing

This section presents the steps performed for transliterating the code switched data to its respective languages.

The given code-mixed dataset portrays the real-time scenario of variations in script changes. Pre-processing is necessary on the text. During pre-processing,

1. To resolve the transliteration variations resulting from script change, we back-transliterated the script to the native language. Our code-mixed datasets have the matrix and embedded languages, where the embedded language is mostly English. Firstly, we used the NLTK[1] English word corpus to detect if the word is in English or not. We used google trans API[2] to detect the word's language id. Later, we back transliterated the non-English word to its native script using a deep transliteration engine[3].

3.2 Background - BERT

Bi-directional Encoder Representations with Transformers (BERT) (Devlin et al., 2018) is a transformer encoder stack trained on the large corpora. BERT uses a transformer architecture (Vaswani et al., 2017). The transformer architecture consists of a series of multi-headed attention, point-wise feed-forward layers with layer normalization to learn contextual relations between words (or subwords) in a text. In our approach, we used a small version of the pre-trained multilingual BERT model called *bert-base-multilingual-cased* obtained from the transformers library. The pre-trained multilingual BERT models are trained on a large multilingual Wikipedia and book corpus. They capture a better semantic representation of words in a text. As the pre-trained model is trained on generic corpora, we need to fine-tune the model for our tasks. During fine-tuning, the pre-trained BERT model parameters are updated.

3.3 Convolutional Model

The convolutional neural network (CNN) model is made of convolutional layers. In short, a convolutional layer uses filters. These filters have a width for processing text. If a filter has a width of 3, then it can see three consecutive tokens. It has many filters, and these filters will slide across the sequence, from beginning to the end, looking at all three consecutive tokens at a time. These filters will learn to extract a different feature from the text. This feature extraction will then be used by the model - potentially as input to another convolutional layer.

Similar to the BERT model, the convolutional model has a positional embedding layer to remember the sequence. The token and positional embeddings are element-wise summed together to get a vector containing information about the token and its position within the sequence. It is followed by a linear layer that transforms the embedding vector into a vector with the required hidden dimension size. The next step is to pass this hidden vector into convolutional blocks. In convolutional blocks, the input sequence is padding such that the length of the input sequence and output sequence should be equal. A Special activation function called gated linear units (GLU) (Dauphin et al., 2017) is used after convolutions. The GLUs have gating mechanisms (similar to Bi-LSTMs (Hochreiter and Schmidhuber, 1997)) contained within the activation function. After passing through the GLU activation, each token's hidden dimension size is the same as it was when it entered the convolutional blocks. Finally, residual connections are applied to solve the vanishing gradients problem, and linear transformations are done to match the dimensions. This process is repeated for N convolutional blocks.

CNN model is formulated as,

$$h_i^l = v\left(\mathbf{W}^l\left[h_{i-k/2}^{l-1}, \ldots, h_{i+k/2}^{l-1}\right] + \mathbf{b}_w^l\right) + h_i^{l-1} \quad (1)$$

Where h_i^l is the output of the i^{th} sequence in l^{th} block. v is the GLU activation function. $\left[h_{i-k/2}^{l-1}, \ldots, h_{i+k/2}^{l-1}\right]$ are convolutional transformations of previous layer, $\mathbf{W}^l$ and $\mathbf{b}_w^l$ are learnable parameters and h_i^{l-1} is the residual output from the previous layer.

After performing the convolutions, Finally, we compute a distribution over the T possible LI or POS tags by transforming the top convolutional encoder output h^L via a linear layer with weights W_{s_2} and bias b_{s_2}.

$$\mathbf{y}_t^{tag} = \text{softmax}\left(\mathbf{W}_{s_2}\mathbf{h}_t^L + \mathbf{b}_{s_2}\right) \quad (2)$$

$$\mathbf{O}_t^{tag} = \text{argmax}\left(\mathbf{y}_t^{tag}\right) \quad (3)$$

[1] https://www.nltk.org/

[2] https://pypi.org/project/googletrans/

[3] https://pypi.org/project/ai4bharat-transliteration/

3.4 Joint Learning Framework

In sequence tagging, we have to learn a function $f : x \rightarrow y$ that maps an input sequence x to the corresponding label sequence y. We want to find the best label sequence y given an input sequence x that maximizes the probability (p) of the sequence given the label.

$$\hat{y} = \underset{y}{argmax}\ p(y|x)$$

In the multilingual code-mixed scenario, if the POS tagging and LI tasks are trained together, joint learning will help the model learn better data representation, improving the tagging accuracy. Thus, we propose a joint training model. The LI and POS tagging tasks can be viewed as conditionally independent in the joint learning scenario, given the transformer encoder parameter values. To jointly train the POS tagging and LI models, the objective is formulated as,

$$p(y^{lang}, y^{pos}|x) = \prod_{t=1}^{T} p(y^{lang}|x, E)p(y^{pos}|x, E)$$

Where $p(y^{lang}, y^{pos}|x)$ is the conditional probability of learning the joint task given the input word sequence, and E stands for encoder parameter values.

The negative logarithm of the above equation gives us the loss function:

$$L = -log \prod_{t=1}^{T} p(y^{lang}|x, E)p(y^{pos}|x, E)$$

$$= -log \prod_{t=1}^{T} p(y^{lang}|x, E) - log \prod_{t=1}^{T} p(y^{pos}|x, E)$$

Which can be further written as

$$L = L^{lang} + L^{pos} \quad (4)$$

If we apply a joint loss function for both the models, we can learn a better POS tagging and LI model for the CM scenario.

4 Experiments

This section evaluates the above method with the individual POS tagging and language identification models on the CM dataset.

Comment	Language id	POS tag
@buttmona098	univ	@
@HamzaIdrees	univ	@
accha	hi	N_NNP
topic	en	JJ
change	en	N_NN
karo	hi	V_VM

Table 1: A example of language and POS tagged Hindi-English code-mixed sentence obtained from the corpus. Where "hi" refers to the Hindi language and "en" reference to the English language

4.1 Dataset

The dataset used for the LI and POS tagging task on the CM language is obtained from the ICON shared task [4]. The CM social media text consists of 3 languages (Bengali, Hindi, and Telugu with mixed English words). The sentences are in roman script. Each word is labeled with its corresponding language label and POS tag. The total number of CM sentences in this dataset is 9212. Two types of POS tagging schemas are used in the dataset. One is a coarse-grained (CR) tagset that used google's universal POS tag-set (Petrov et al., 2011) and the other is fine-grained tagset (FI) with an extended tag-list related to social media text (Gimpel et al., 2010; Owoputi et al., 2013). We have used both POS tagsets in our experiments. The initial dataset contained errors in tagsets, which were addressed in the article (Bhattu et al., 2020b). We have used the revised tagset in our paper obtained from (Bhattu et al., 2020b). An tagged Hindi-English sentence is given in the table 1.

4.2 Baselines

We compared the performance of our model with the related works on the CM data. The acronym "(I)" refers to the individual model, and "(J)" refers to the joint model.

CRF (I): A CRF model is used for language identification on the CM data. The features set defined in (Gundapu and Mamidi, 2018) is used while training the CRF model.

FCRF (J): A factorial CRF model with joint learning (Barman et al., 2016) for LI and POS tasks.

Bi-LSTM CRF (I): This is the most commonly used model for the LI and POS tasks. It uses Bi-

[4] http://www.amitavadas.com/Code-Mixing.html

LSTM and CRF consecutively for POS tagging the CM text (Aguilar and Solorio, 2019; Bhattu et al., 2020a).

Bi-LSTM CRF (J): A Bi-LSTM CRF model that is trained for the task of joint learning (Soto and Hirschberg, 2018).

BERT (I): Pre-trained multilingual BERT (Devlin et al., 2018) is used for the given LI and POS tasks.

BERT (J): A joint BERT model (Chen et al., 2019) is used for CM language identification and POS tagging.

4.3 Implementation

We implement the proposed model as follows. The input text is pre-processed. The pre-processed text is given to the BERT model. The BERT encodes the input text. The encoded input is given to the convolutional model. The encoded output is given to the 2 multi-layer perceptron models (MLP), each one for the two tasks LI and POS tagging. Then we aggregate the loss of the two models. The loss is then propagated backward, and the model optimizes to minimize the losses of both the NN models.

The proposed model is trained using Adam optimizer with cross-entropy loss obtained from the joint model. The hyperparameters are: Optimization $(\alpha) = 0.001$, dropout probability is 0.25. The number of epochs used is 10. All the deep learning models are implemented in python 3.6 using the PyTorch and the Torchtext libraries. We used the NVIDIA RTX 2070 graphics card with an $8GB$ GPU memory. We have used python-crfsuite[5] and pytorch-crf[6] for the CRF model, and the BERT model is obtained from the transformers[7] library.

4.4 Performance

We compared our model with all the baselines and the results are tabulated in tables 2,3 and 4. The proposed model has shown an improvement in the language identification and the POS tagging tasks compared to the baseline models (CRF and Bi-LSTM CRF). Using joint learning, The joint model has seen a further improvement when compared to the individual models.

Even the BERT model has proved better than the other models because of its state-of-the-art transformer architecture. We have observed that the transformer architecture helped the model to learn the tags that have a low frequency and gave better recall for the low-frequency tags. We observed the same in our convolutional seq2seq model.

Our model, the BERT with the convolutional seq2seq architecture, formed a meta-learning approach on the code-mixed text. The BERT model learned the better representation of the data. The kernels used in the convolutional seq2seq model helped the model consider the previous tagged information while predicting the current tag. We had observed that the suffixes are classified correctly when our approach was used.

For Language Identification, we have seen a considerable difference in joint learning and pos tagging when compared to the CRF model on the Telugu-English dataset. We have observed that most of the misclassified *te(Telugu-word)* and *en(English-word)* are correctly classified with joint learning. Even the *acro(acronym)* words which were of low frequency, were better identified with our approach. Some *mixed* words, i.e., with intra level code-mixing, are present in the data. These have undergone sub-word tokenization and were unidentifiable by the BERT model. As the CNN model uses the filter of size 3, the previous two tokens will be considered while predicting the tag of the current token. The CNN model and the pre-processing step helped the model to detect the *mixed* words correctly, which further improved the performance of our model.

In fine-grained and coarse-grained POS tagging, the major shift in joint-learning and individual task learning is observed in the baseline CRF model on the Bengali-English coarse-grained dataset. We have observed that the incorrect tagging of G_N as G_V or vice versa has decreased when joint learning is used for the POS tagging scenario.

We have also observed that our approach performed better in tagging G_N and G_V words when compared to the CRF and Bi-LSTM CRF models. It resulted in improved accuracy. It is due to its state-of-the-art multi-headed attention feature used in this model.

In the code-mixing scenario, the word's POS tag depends on the following factors: how the word is used in the CM sentence, its multilingual feature, and its context. With the help of joint learning, which considered the grammatical structure and multilingual nature of the word, the incorrect classification problem of tagging G_N as G_V and

[5] https://pypi.org/project/python-crfsuite/
[6] https://pytorch-crf.readthedocs.io/en/stable/
[7] https://pypi.org/project/transformers/

LI	Be-En		Hi-En		Te-En	
	macro-F1	**Acc**	**macro-F1**	**Acc**	**macro-F1**	**Acc**
CRF (I)	55.16	74.15	63.14	87.17	70.12	83.51
CRF (J)	55.29	74.80	63.46	88.54	70.27	85.77
Bi-LSTM + CRF (I)	56.49	81.87	63.40	88.85	70.38	87.23
Bi-LSTM + CRF (J)	56.80	82.47	63.56	89.15	70.41	87.91
mBERT (I)	58.17	89.81	63.31	95.66	70.45	87.70
mBERT (J)	58.20	90.23	63.51	96.19	70.48	88.61
pre-process + mBERT + CNN (I)	**58.21**	**91.18**	**63.63**	**97.61**	**70.51**	**89.13**
pre-process + mBERT + CNN (J)	**58.25**	**91.21**	**63.71**	**97.69**	**70.52**	**89.65**

Table 2: Macro-F1 and Accuracy metric for Language Identification on CM social media data (Be stands for Bengali, En for English, Hi for Hindi, Te for Telugu)

POS (CR)	Be-En		Hi-En		Te-En	
	macro-F1	**Acc**	**macro-F1**	**Acc**	**macro-F1**	**Acc**
CRF (I)	52.76	65.93	56.38	71.24	60.92	68.24
CRF (J)	52.85	69.69	56.57	71.90	60.93	70.41
Bi-LSTM + CRF (I)	53.37	78.56	57.23	86.62	60.26	87.70
Bi-LSTM + CRF (J)	53.83	79.43	57.51	87.72	60.16	88.12
mBERT (I)	54.17	78.81	57.61	86.66	60.45	87.87
mBERT (J)	54.20	79.12	57.71	87.19	60.48	89.61
pre-process + mBERT + CNN (I)	**54.83**	**79.85**	**57.88**	**88.76**	**60.91**	**90.31**
pre-process + mBERT + CNN (J)	**54.92**	**80.23**	**57.90**	**89.79**	**61.52**	**91.65**

Table 3: Macro-F1 and Accuracy metric for coarse-grained (CR) POS tagging on CM social media data (Be stands for Bengali, En for English, Hi for Hindi, Te for Telugu)

POS (FN)	Be-En		Hi-En		Te-En	
	macro-F1	**Acc**	**macro-F1**	**Acc**	**macro-F1**	**Acc**
CRF (I)	45.24	56.19	45.69	57.13	45.14	64.48
CRF (J)	45.63	68.17	47.17	58.56	45.90	66.17
Bi-LSTM + CRF (I)	47.24	79.97	48.63	79.45	47.55	87.14
Bi-LSTM + CRF (J)	47.23	79.43	49.51	81.72	47.16	89.12
mBERT (I)	48.43	78.21	50.13	81.35	48.45	87.87
mBERT (J)	48.45	78.47	50.48	82.17	48.90	88.64
pre-process + mBERT + CNN (I)	**48.51**	**78.41**	**50.17**	**81.44**	**48.67**	**88.30**
pre-process + mBERT + CNN (J)	**48.47**	**78.67**	**50.78**	**81.49**	**48.71**	**89.18**

Table 4: Macro-F1 and Accuracy metric for Fine-grained (FN) POS tagging on CM social media data (Be stands for Bengali, En for English, Hi for Hindi, Te for Telugu)

vice-versa is reduced.

We have observed low macro-F1 scores in our tagging models. It is due to the presence of distinct tags with low frequency and was insufficient to be trained by the existing models. These tags compromised the F1 score on the given data.

The experiments conducted on various models show that the joint learning model achieves improved POS tagging and LI in the code mixed scenario.

5 Conclusion

In this paper, we presented the joint BERT with the CNN model for POS tagging and LI. The joint learning model allowed the POS tagging and LI to be conditioned on each other to achieve better processing of code-mixed text. We tested our model with individual tasks. The results prove that our model achieves better metrics when compared to individual models. Such relations can effectively achieve better sentence-level semantic representa-

tion due to such diverse learning scope.

Recently meta embedding representations that include both the pre-trained embeddings and domain-specific fine-tuned embeddings are achieving great results in the field of NLP. The joint learning with meta embeddings is left as future scope.

References

Gustavo Aguilar and Thamar Solorio. 2019. From english to code-switching: Transfer learning with strong morphological clues. *arXiv preprint arXiv:1909.05158*.

Kelsey Ball and Dan Garrette. 2018. 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*, pages 3084–3089, Brussels, Belgium. Association for Computational Linguistics.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*, pages 13–23.

Utsab Barman, Joachim Wagner, and Jennifer Foster. 2016. Part-of-speech tagging of code-mixed social media content: Pipeline, stacking and joint modelling. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 30–39.

S. Nagesh Bhattu, Satya Krishna Nunna, D. V. L. N. Somayajulu, and Binay Pradhan. 2020a. Improving code-mixed pos tagging using code-mixed embeddings. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(4).

S Nagesh Bhattu, Satya Krishna Nunna, Durvasula VLN Somayajulu, and Binay Pradhan. 2020b. Improving code-mixed pos tagging using code-mixed embeddings. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(4):1–31.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Suman Dowlagar and Radhika Mamidi. 2021. Multilingual pre-trained transformers and convolutional nn classification models for technical domain identification. *arXiv preprint arXiv:2101.09012*.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2010. Part-of-speech tagging for twitter: Annotation, features, and experiments. Technical report, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

Sunil Gundapu and Radhika Mamidi. 2018. Word level language identification in english telugu code mixed data. In *PACLIC*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. Association for Computational Linguistics.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. Gluecos: An evaluation benchmark for code-switched nlp. *arXiv preprint arXiv:2004.12376*.

Deepthi Mave, Suraj Maharjan, and Thamar Solorio. 2018. Language identification and analysis of code-switched social media text. In *Proceedings of the third workshop on computational approaches to linguistic code-switching*, pages 51–61.

Carol Myers-Scotton. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380–390.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

Victor Soto and Julia Hirschberg. 2018. 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*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.