Classification of Code-Mixed Text Using Capsule Networks

Shanaka Chaturanga, Surangika Ranathunga
Department of Computer Science and Engineering,
University of Moratuwa,
Katubedda 10400, Sri Lanka.
[shanaka.chathuranga.19, surangika]@cse.mrt.ac.lk

Abstract

A major challenge in analysing social media data belonging to languages that use non-English script is its code-mixed nature. Recent research has presented state-of-the-art contextual embedding models (both monolingual s.a. BERT and multilingual s.a. XLM-R) as a promising approach. In this paper, we show that the performance of such embedding models depends on multiple factors, such as the level of code-mixing in the dataset, and the size of the training dataset. We empirically show that a newly introduced Capsule+biGRU classifier could outperform a classifier built on the English-BERT as well as XLM-R just with a training dataset of about 6500 samples for the Sinhala-English code-mixed data.

1 Introduction

Social media has become very popular among people across the world during the last decade, mainly due to the popularity of smart mobile phones. For low-resource languages, this social media data has become a major source of text in building Natural Language Processing (NLP) applications. Most of the content in social media tends to be informal. When the users are (at least to a certain degree) multilingual, the informal content they publish in social media tends to be code-mixed. Code-mixed data is a result of code-switching, which denotes a shift from one language to another within a single utterance (Sitaram et al., 2019).

Recent work on text classification with code-mixed data has used contextual embedding models such as BERT (Devlin et al., 2019a), and their multilingual versions, such as mBERT or XLM-R (Aguilar et al., 2020; Kumar et al., 2020). However, using these pre-trained models for code-mixed text classification, in particular the domain-specific data in low-resource languages poses many challenges. Contextual embedding models such as BERT need large volumes of monolingual data to train. On the other hand, not every language is included in the pre-trained multilingual models, and low-resource languages are underrepresented in those models (due to the smaller amounts of low-resource language data used in contrast to high-resource languages when training these models). In fact, there is a line of research that has shown even simple classifiers such as Logistic Regression proving to be more effective than the multilingual embedding models (Chakravarthi et al., 2020).

In this paper, we empirically show that the success of contextual embedding models on code-mixed text classification depends on multiple factors, and they can indeed be sub-optimal compared to text classification based on neural models other than transformer based ones.

We selected Sinhala and Malayalam, which are low-resource languages. In the recent language categorization by Joshi et al. (2020), Sinhala belongs to class 0 (i.e. it has exceptionally limited resources). Malayalam is categorised as 1, (i.e. it has some unlabelled data, however collecting labelled data is challenging).

A food recipe dataset with about 3000 samples (Kazhuparambil and Kaushik, 2020) was used as the Malayalam-English code-mixed data. For Sinhala-English code-mixed data, a corpus of 10000 user comments in the domain of telecommunication was annotated with two types of information: aspects related to the telecommunication domain, and overall sentiment of the comment.

We fine-tuned English-BERT and XLM-R (Conneau et al., 2019) models on both datasets. Then we implemented a novel Capsule+biGRU network for the same tasks. Results show that the Capsule+biGRU model consistently outperforms English-BERT and XLM-R models for the Sinhala-English dataset that had more data and less code-mixing complexity than the Malayalam-English dataset. With this, we establish the argument that the performance of contextual embedding mod-

Proceedings of Recent Advances in Natural Language Processing, pages 256–263
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_030

els depends on multiple factors such as the code-mixing level, size of the dataset used to train the contextual embedding models, and the size of the dataset used in fine-tuning. Further experiments with the Sinhala-English dataset showed that this Capsule+biGRU model is superior to recurrent models as well. The annotated dataset, as well as our code are publicly released [1]

2 Related Work

2.1 Deep Learning based Text Classification

Aspect identification and sentiment classification tasks used in this paper are essentially text classification problems. Thus, without any loss of generality, in this section we look at Deep Learning solutions applied for text classification. Minaee et al. (2021) identified several Deep Learning techniques for supervised text classification. The simplest technique is the Feed Forward networks, which treats the input text as a bag of words.

Subsequently introduced Recurrent Neural Models have the ability to capture the sequential dependencies between words. Long-Short Term Memory Networks (LSTMs), and Gated Recurrent Units (GRUs) were introduced to solve some of the shortcomings of the RNN models. LSTMs, in particular bi-LSTMs have been very commonly used. There have been several LSTM variants employed in text classification such as tree-LSTM (Tai et al., 2015), multi-timescale LSTM (Liu et al., 2015), and sentence-state LSTM (Zhang et al., 2018). It is common to use pre-trained word embedding models such as Word2Vec or fastText to be used as the input representation of these recurrent models. Attention mechanism is employed on top of architectures such as GRUs (Yang et al., 2016), or LSTMs (Liu et al., 2016).

Convolutional Neural networks (CNNs) is another model used for text classification. Similar to LSTMs, different CNN variants such as character-level CNNs (Zhang et al., 2015), and multi-layer CNNs (Pang et al., 2016) have been employed.

However, CNN has a problem of information loss with respect to its pooling operation. More recently, capsule networks were introduced to address this problem, and have been reported to outperform CNN based text classification systems (Yang et al., 2019), as well as those based on recurrent models like LSTMs (Senevirathne et al., 2020).

Recently introduced Transformers (Vaswani et al., 2017) are now being commonly used to build extremely large language models (also known as contextual embedding models). The most popular contextual embedding model is BERT (Devlin et al., 2019a), and there have been subsequent improvements to it. Text classification with pre-trained language models has now become the state-of-the-art for Text Classification (Bao et al., 2020).

2.2 Classifying Code-Mixed Data

Research on code-mixed data spans across tasks such as language identification (Gundapu and Mamidi, 2018), Part of Speech tagging (Vyas et al., 2014), speech recognition (Shah et al., 2020) and text classification. In this discussion, we only focus on text classification of code-mixed data.

Out of the aforementioned Deep Learning techniques, code-mixed data classification has been mainly implemented using LSTMs, and contextual embedding models. English-BERT was employed because in code-mixed data, foreign language text is written in English script. Moreover, multilingual contextual embedding models s.a. LASER (Artetxe and Schwenk, 2019), mBERT (Devlin et al., 2019b) and XLM-R have been employed. Related research reported mixed observations on the performance of these techniques. Aguilar et al. (2020) showed that mBERT performs better than biLSTM that has one-hot vector representation as the input, while Yadav and Chakraborty (2020) showed that an LSTM trained with domain-specific embeddings as input representations performed better than LASER. Interestingly, some research reported that Machine Learning algorithms such Logistic Regression and Random Forest were able to outperform BERT and even mBERT (Chakravarthi et al., 2020; Javdan et al., 2020).

Compared to monolingual text classification, the number of code-mixed datasets is limited. The most notable one is the Spanish-English and Hindi-English code-mixed datasets released for the SemEval 2020 task (Javdan et al., 2020). Other than that, there are code-mixed datasets available between Malayalam-English (Kazhuparambil and Kaushik, 2020), and Tamil-English (Chakravarthi et al., 2020). Interestingly, other than the Spanish-English dataset, all the other datasets we identified involve Indic languages.

[1] https://github.com/shanakaChathu/ABSA

Figure 1: Class distribution of Malayalam-English Dataset

Text	Class
Thankyou for lakshmi nair vlogs	Suggestions and queries
That chopping was ohh veenechi thanku soo much	About the receipe
Super Tea cake Veena!!! Wl surely try	About the receipe

Table 1: Sample of Code-mixed Malayalam-English Dataset

3 Datasets

3.1 Malayalam-English Dataset

The Malayalam-English dataset was obtained from A food recipe dataset with about 3000 samples Kazhuparambil and Kaushik (2020). Figure 1 and the Table 1 show the class distribution and few samples of the Malayalam-English dataset, respectively. Each comment is tagged with one of the seven classes. Most of the comments in the dataset belong to the undefined class. Comments written in the Malayalam characters were removed from the dataset, and the resulting dataset had 3434 records. Thus the dataset has English words, as well as Malayalam words written in English script. Overall, about 25% of the corpus is English words.

3.2 Sinhala-English Dataset

This dataset was newly created by us. Telecommunication domain has been identified as a low-resource domain. We are not aware of any dataset or research that considered text data in this domain. This dataset was annotated for the following two text classification tasks.

- Document-level sentiment classification, where each user comment is annotated with its sentiment - positive, negative, and neutral.
- Aspect extraction, where each comment is annotated with the aspect term it refers to. Aspects are specific to the telecommunication domain, as discussed below.

Dataset	Size	Average Comment Length
Si-En	10006	Positive: 61 Negative:62 Neutral : 85
Ma-En	4291	Gratitude : 55 About The Recipe: 46 About the Video: 44 Praising:60 Hybrid:91 Undefined:59 Suggestions and Queries:69

Table 2: Dataset Details

A mini-survey was conducted with the help of 150 users of the telecommunication companies through the social media. First, a detailed list of aspects was identified by analysing user comments. An online form was distributed among social media users and they were asked to select the most important aspects from the aspect list. After that, the highly rated set of aspects was selected as the aspects to be annotated in the corpus. Six aspects, namely network, billing or price, package, customer service, data, and service or product were identified as the final aspects using that mini-survey.

Data was extracted from public forums in FaceBook. All these FaceBook pages can be accessed without logging into FaceBook, and are indexed by search engines. All the company names and people names included in the comments were manually removed from the dataset.
Two annotators were employed for the aspect and sentiment annotation. 10000 comments were annotated with the aspects and the sentiment. The sentiment distribution and aspect distribution are shown in the Figure 2 and Figure 3, respectively. Also a sample of the telco data is shown in Table 3. This dataset only contains English and Sinhala written in the English script. Overall, about 10% of the corpus refers to English words. Comment length related statistics are available in the Table 2. The inter-annotator agreement was calculated using Cohen's kappa statistics. It is 0.61 for aspect identification while 0.67 for sentiment classification.

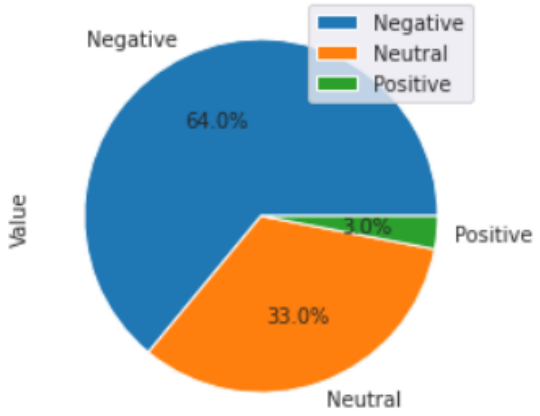

Figure 2: Sentiment Distribution of Telco Dataset

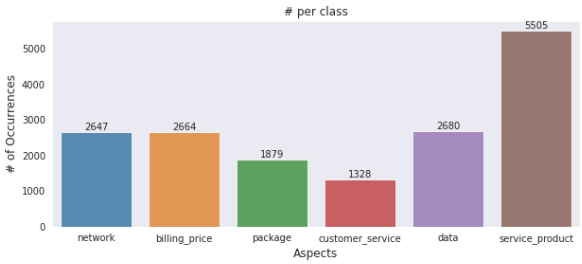

Figure 3: Aspect Distribution

4 Methodology

4.1 Pre-Processing

The Sinhala-English dataset was pre-processed to reduce the noise. Initially, punctuation characters were removed from the text. After that URLs, mobile phone numbers, and Emails were removed from the text. Also, social media comments contain many emojis in the comments. Those were converted to the word format. The Hashmark was removed from the hashtag as another pre-processing step. After doing all the above-mentioned steps, words were converted to their lowercase form, and the sentences were tokenized. After that basic text pre-processing steps such as converting to lowercase and stop word removal were done to the both datasets.

Comment	Network	Billing_ price	Customer_ Service	Data	Service_ Products	Package	Sentiment
Tik tok app eka wada na me pack ekata	0	0	0	0	0	1	-1
Unlimited data dunnata godak slow	1	0	0	0	0	0	0

Table 3: Sample of Telco(Si-En code mixed) Dataset

4.2 Classifying Code-Mixed Social Media Data

In the sentiment analysis problem, only one class is predicted from the positive, negative or neutral classes. But in the aspect prediction mode, more than one class may be predicted if that comment contains more than one aspect. Because of that, sentiment analysis problem was resolved as a multi-class classification problem while the aspect prediction problem was resolved as a multi-label classification problem.

Firstly, in order to setup the baselines, we used recurrent deep learning models on the Sinhala-English dataset. These include RNN, LSTM, GRU, and BiLSTM. fastText and Word2Vec models were trained from a raw corpus of 100000 words extracted from the same sources that were used to create the Sinhala-English annotated dataset. These embeddings were used as the input representation of all these neural models.

Secondly, various improvements were carried out on these models as described below:.

- Regularization strategies such as dropout, L1/L2 regularization, and early stopping.
- Integration with CNN models, because CNNs are known to be able to extract more coarse-grained features.
- Stacked models with the aim of extracting rich contextual knowledge from the network's upper layers. These stacked models contain additional higher layers that extract valuable contextual information from both past and future time sequences (Zhao et al., 2018).

Thirdly, a capsule network was implemented. Capsule networks were selected because they have performed better than LSTM,RNN,Bi-LSTM, etc with Sinhala text classification (not on code-mixed data) (Senevirathne et al., 2020). The capsule network was introduced as an upgrade to CNNs, to be used in NLP applications such as text classification (Sabour et al., 2017). The ability to record context level information in its precise sequence using a vector representation of the capsules is a crucial aspect of the capsule architecture. The dynamic routing mechanism of the capsule network is known to overcome drawbacks of CNNs such as high computational cost and information loss caused by the widely utilized max pooling approach. This basic capsule network architecture

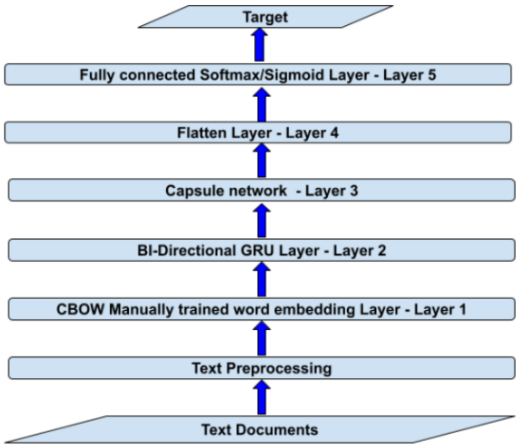

Figure 4: Capsule+biGRU Architecture

was combined with LSTM, GRU, BiLSTM, and biGRU models. However, only the combination with the biGRU model gave better results than the recurrent and CNN models, thus only that result will be reported.

Figure 4 shows the Capsule+biGRU network we employed. According to that, firstly raw comments were pre-processed using the text-processing techniques (see Section 4.1). After that, the above trained CBOW word embeddings were used as the first layer of the neural network followed by the Bi-Directional GRU layer. Output from the GRU layer was used in the capsule layer. Finally, a flatten layer was implemented followed by a fully connected Softmax of Sigmoid layer.

Finally, we experimented with the pre-trained contextual embedding models. We experimented with English-BERT and XLM-R. mBERT was not used because it does not include Sinhala. A text classification layer was added on top of both English-BERT and XLM-R models, and they were fine-tuned with the English-Sinhala training data.

To further establish the performance of Capsule+biGRU with respect to English-BERT and XLM-R, these models were tested on the Malayalam-English dataset. Recurrent and CNN models were not tested, as their performance lagged behind the Capsule+biGRU model for Sinhala-English. BERT-uncased model[2] was used as the pre-trained model in all the experiments.

[2] https://huggingface.co/bert-base-uncased

Parameter	Parameter value
Optimizer	SGD, RMSprop, Adamax, Adagrad
Dropout Rate	0.25,0.50,0.70
GRU Activation	relu, tanh, linear
Number of Capsules	5, 10, 20, 40
Dimension of Capsules	8, 16, 32, 64
GRU Length	16, 64, 128, 256

Table 4: Hyper-Parameters of the Capsule+biGRU Network

Embedding size	Word2Vec (CBOW)	FastText
100	0.835	0.824
200	0.822	0.803
300	0.839	0.818
400	**0.845**	0.817
500	0.838	0.804

Table 5: Word Embedding Results.

5 Experiments and Evaluation

5.1 Experiment Setup

All experiments were carried out using Google Colab and kaggle. Python was used as the main programming language while Keras was used to build deep neural networks. Three-fold cross-validation was used in every experiment. After that, the best model was trained with the hold-out-based method and hyper-parameter tuning was carried out. Hyper-parameters used in the capsule+biGRU model are shown in the Table 4. The dataset was split into train and validation sets with ratios of 5:1 when doing the hold-out based experiments. Accuracy, precision, recall and F1 were reported as the weighted average for each experiment on the cross-validated dataset.

5.2 Word Embedding Models

The first experiment was to identify the best word embedding models for Sinhala. Both Word2Vec (CBOW) and fastText models were tested for 100,200,300,400 and 500 dimensions. 100000 comments extracted from the same dataset were used to build the word embeddings. CNN model was used as the model to find the best word embedding technique and embedding size, as doing this experiment for all the models is not possible.

According to Table 5, Word2Vec (CBOW) word embedding model with 400 embedding size showed the highest weighted F1 score compared to the

Model Type	Model	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)
Sentiment Model	CNN	80.4	81.2	79.9	80.5
	Stacked BiLSTM 3	75.5	75.7	74.5	75.1
	CNN+BiLSTM	79.2	80.1	77.9	79.0
	CNN+GRU	64.4	64.3	64.3	64.3
	CNN+Stacked BiLSTM 2	76.9	77.0	76.5	76.8
	CNN+Stacked BiLSTM 3	76.7	77.4	76.9	76.1
	Capsule +BiGRU	81.9	82.5	81.0	**81.7**
	BERT	73.4	79.2	73.3	71.7
	XLM-R	70.0	77.1	70.0	69.0
Aspect Model	CNN	62.2	84.2	70.6	76.4
	Stacked BiLSTM 3	61.9	82.1	71.8	76.1
	CNN+BiLSTM	59.5	82.6	73.7	77.7
	CNN+GRU	27.45	54.7	34.5	42.0
	CNN+ BiLSTM 2	60.8	81.2	71.0	75.5
	CNN+ BiLSTM 3	61.9	82.1	71.8	76.1
	Capsule + BiGRU	89.8	83.7	79.1	**81.1**
	BERT	54.1	80.7	81.1	80.6
	XLM-R	52.4	81.7	79.1	79.4

Table 6: Results of Sentiment Model and Aspect Model for Sinhala-English (3-Fold cross validation).

Model	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)
BERT	66.8	67.2	66.8	**66.9**
XLM-R	64.82	65.6	64.8	64.9
Capsule+biGRU	53.1	67.1	44.3	52.6

Table 7: Text classification results of Malayalam-English Dataset (3-Fold cross validation).

other model. The aspect prediction model also suggested the same thing. Because of that, CBOW with 400 embedding size was used in all experiments.

Table 6 and Table 7 show the results for the two Sinhala-English tasks and the the Malayalam-English task, respectively. Note that the Malayalam-English dataset was used only to compare the Capsule+biGRU against English-BERT and XLM-R models.

In the sentiment classification task, the Capsule+biGRU model significantly outperforms English-BERT and XLM-R based solutions. However, the gain in the aspect identification task is not that significant. The result with the Malayalam-English dataset is quite the opposite- the capsule+biGRU model significantly lags behind English-BERT and XLM-R models.

We can think of multiple reasons for this observation. First and foremost, Sinhala-English dataset was much larger than the Malayalam-English dataset. We believe the number of training samples in the latter dataset was not sufficient to train the Capsule+biGRU model. In contrast, the pre-trained models could cope with this lack of data. On the other hand, the Malayalam-English dataset had a much higher number of English words compared to the Sinhala-English dataset, which could have been an advantage for the English-BERT model, as well as the XLM-R model that has a significant presence of English. Another reason could be the complexity of the Malayalam-English dataset. Despite the task, XLM-R is consistently lagging marginally behind English-BERT. We attribute this observation to the fact that Sinhala and Malayalam being underrepresented in the XLM-R model. Though we do not know exact size of the commoncrawl corpus used to train the XLM-R model, according to the latest commoncrawl statistics[3], Sinhala and Malayalam representation was just 0.0070% and 0.0211 %, respectively.

6 Conclusion

The objective of this research was to critically analyse the performance of English-BERT and XLM-R models for classifying code-mixed data. We identified that the performance of these models depends on factors such as the size and composition of the code-mixed data. We were able to introduce a novel Capsule+biGRU model that could outperform the

[3] https://commoncrawl.org/2021/05/may-2021-crawl-archive-now-available/

English-BERT and XLM-R models with a moderate dataset of Sinhala-English 10000 comments (Note in 3-fold cross-validation, about 6600 samples are used for training). This result suggests that, at least for text classification on code-mixed data that involves extremely low-resource languages that are under-represented in the large multilingual embedding models, traditional Deep Learning solutions are still a viability. This research can be considered as one of the very few works that comparatively analysed the performance of these different techniques for code-mixed data with respect to multiple factors and languages. Furthermore, this research publicly released a code-mixed dataset that can be used for two text classification tasks for the extremely low resource language Sinhala. We believe that further research should be conducted with respect to more languages in order to properly determine the impact of the aforementioned factors on classification of code-mixed text.

Acknowledgments

This publication was funded by a Senate Research Committee (SRC) Grant of University of Moratuwa, Sri Lanka.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. Lince: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1803–1813.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020. Corpus creation for sentiment analysis in code-mixed tamil-english text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sunil Gundapu and Radhika Mamidi. 2018. Word level language identification in english telugu code mixed data. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*.

Soroush Javdan, Behrouz Minaei-Bidgoli, et al. 2020. Iust at semeval-2020 task 9: Sentiment analysis for code-mixed social media text using deep neural networks and linear baselines. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1270–1275.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Subramaniam Kazhuparambil and Abhishek Kaushik. 2020. Cooking is all about people: Comment classification on cookery channels using bert and classification models (malayalam-english mix-code). *arXiv preprint arXiv:2007.04249*.

Ayush Kumar, Harsh Agarwal, Keshav Bansal, and Ashutosh Modi. 2020. Baksa at semeval-2020 task 9: Bolstering cnn with self-attention for sentiment analysis of code mixed text. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1221–1226.

Pengfei Liu, Xipeng Qiu, Xinchi Chen, Shiyu Wu, and Xuan-Jing Huang. 2015. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2326–2335.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2793–2799.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*.

Lahiru Senevirathne, Piyumal Demotte, Binod Karunanayake, Udyogi Munasinghe, and Surangika Ranathunga. 2020. Sentiment analysis for sinhala language using deep learning techniques. *arXiv preprint arXiv:2011.07280*.

Sanket Shah, Basil Abraham, Sunayana Sitaram, Vikas Joshi, et al. 2020. Learning to recognize code-switched speech without forgetting monolingual speech recognition. *arXiv preprint arXiv:2006.00782*.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979.

Siddharth Yadav and Tanmoy Chakraborty. 2020. Unsupervised sentiment analysis for code-mixed data. *arXiv preprint arXiv:2001.11384*.

Min Yang, Wei Zhao, Lei Chen, Qiang Qu, Zhou Zhao, and Ying Shen. 2019. Investigating the transferring capability of capsule networks for text classification. *Neural Networks*, 118:247–261.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 649–657.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state lstm for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.