Sentiment Analysis in Code-Mixed Telugu-English Text with Unsupervised Data Normalization

by

Siva Subrahamanyam, Preetham Reddy Sathineni, 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

Sentiment Analysis in Code-Mixed Telugu-English Text with Unsupervised Data Normalization

Subrahamanyam Varma, Preetham Sathineni, Radhika Mamidi
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, Telangana, India
{siva.subrahamanyam, preethamreddy.s}@research.iiit.ac.in
radhika.mamidi@iiit.ac.in

Abstract

In a multilingual society, people communicate in more than one language, leading to Code-Mixed data. Sentimental analysis on Code-Mixed Telugu-English Text (CMTET) poses unique challenges. The unstructured nature of the Code-Mixed Data is due to the informal language, informal transliterations, and spelling errors. In this paper, we introduce an annotated dataset for Sentiment Analysis in CMTET. Also, we report an accuracy of 80.22% on this dataset using novel unsupervised data normalization with a Multilayer Perceptron (MLP) model. This proposed data normalization technique can be extended to any NLP task involving CMTET. Further, we report an increase of 2.53% accuracy due to this data normalization approach in our best model.

1 Introduction

In recent times, huge volumes of data are being generated on social media and online blogging. The usage of multiple languages in day-to-day conversations and the minimal linguistic restrictions on the online content lead to the generation of Code-Mixed data. Code-Mixing or Code-Switching is the usage of two or more languages in a single sentence or a conversation. The generated code-mixed data can be used to extract potential knowledge like emotion, news or information.

Sentimental analysis on Code-Mix social media data helps us to understand the underlying sentiment of the phrase or sentence, which can have many practical use cases in the real world. For example, it can be used to understand the sentiment of restaurant or movie reviews.

Code-Mixed Data can be easily extracted from various online platforms like social media and blogs using APIs and various web-scraping tools. In spite of having huge amounts of data, performing sentiment analysis on it is still challenging because of its unstructured and noisy nature (Arora and Kansal (2019), Gautam and Yadav (2014), Barik et al. (2019)). Hence, this requires robust preprocessing techniques to normalize this unstructured data.

In this paper, we have performed sentimental analysis on Code-Mixed Telugu-English Text (CMTET) with an unsupervised data normalization. We analyse and classify each sentence into three sentiments, positive, negative and neutral. To the best of our knowledge this is the first work to propose a framework to perform sentiment analysis on CMTET. The reason for choosing English as our secondary language for the Code-Mixed data is because we observed that most of the Telugu data available in social media is very often Code-Mixed with English.

The rest of the paper is organized as follows. In section 2, we discuss the existing Sentiment Analysis approaches in other Code-Mixed languages. Section 3 introduces the dataset and describes the methodology involved in preprocessing and annotating this dataset. In section 4, we discuss in detail various challenges faced in performing sentiment analysis in CMTET. In section 5, we discuss about the proposed data normalization technique for CMTET. In section 6, we explain the process of feature extraction and sentiment classification. In Section 7, we show the results of the proposed methods and make a comparative study on the accuracy of all those models. In section 8, we discuss the problems faced by the models in predicting the sentiment. Section 9 concludes the paper with the summary and scope for future work.

2 Related Work

For the task of Sentiment Analysis on Code-Mixed Hindi-English text, Sharma et al. (2015) used a lexicon-based approach on the FIRE 2013 (Roy et al., 2013) and FIRE 2014 datasets. The dataset consisted of the user comments from public Facebook pages of two of the most popular celebrities. Joshi et al. (2016) introduced a new dataset and used subword-LSTM to address the noisy nature of the text and reported an improvement of 18% on the baseline. Choudhary et al. (2018) proposed clustering of Code-Mixed word variations with skip-gram vectors based on similarity. They also used contrastive learning and projected the Code-Mixed sentences into a common sentiment space using shared parameters. Sukhpreet Kaur (2021) used attention based models with word-level, sub-word level and character-level representations of the sentences, and reported that Bi-LSTM performed the best.

Chakravarthi et al. (2020) had presented an annotated code-mixed corpus in Malayalam-English language of Youtube comments for the task of Sentiment Analysis. Kalaivani and Thenmozhi (2020) performed Sentiment Analysis in Code-Mixed Malayalam-English and Tamil-English text using AWD-LSTM(Merity et al., 2017) and reported a weighted F1-Score of 0.6 on both the datasets. Mishra et al. (2018) proposed an ensemble model to perform sentiment analysis in Hindi - English and Bengali - Hindi - English corpus. The ensemble model is built on linear SVM, logistic regression and random forest based on a voting classifier.

Sabri et al. (2021) created a Persian-English code mixed data corpus from tweets. They also proposed a Bi-LSTM based ensemble model which uses BERT embeddings and translation models to learn sentiment of tweets. Yadav and Chakraborty (2020) proposed a zero-shot approach to solve Sentiment Analysis task on Code-Mixed Spanish-English text. They used multilingual and crosslingual embeddings to transfer knowledge from monolongual text to Code-Mixed text. They reported an increase of 3% in accuracy over the previous state-of-the-art.

In the task of Sentiment Analysis on Social Media text, there is a lot of research on using different models to address the problem of its noisy nature. Our work differs in this context, where, we propose an unsupervised approach for normalizing CMTET. And to the best of our knowledge, this is the first work of Sentiment Analysis in CMTET.

3 Dataset

In this paper, we introduce a new dataset for sentiment analysis in CMTET. The methodology consists of three main phases: data collection, data cleaning, and data annotation. The below sections describe each phase in detail.

3.1 Data Collection

We have identified a few Twitter users and regional moview review Youtube videos. We have observed that these Twitter users tweet in CMTET on different aspects such as Movies and Sports. Also, the identified Youtube videos contain user comments expressing their sentiment of the movie or the video in CMTET. We used Twitter public streaming API [1] and Youtube Comments API[2] to collect this data.

3.2 Data Cleaning

We removed irrelevant text such as URLs and markup text, using regex-based[3] pattern matching to ensure basic data quality. Then, with the help of NLTK Tokenizer[4], we tokenized each sentence into words. We then removed sentences containing less than five words, as we observed that these sentences are noisy and hardly contain any information.

3.3 Data Annotation

We adopted the three-class classification for sentiment of the sentences, i.e., positive, negative, and neutral (Koppel and Schler, 2006). The objective of this phase is to annotate each sentence. We also annotated this data at word-level with their language using the language tags, i.e., English (EN), Telugu (TE), Named Entities (NE), and Universal(UNIV).

[1] https://developer.twitter.com/en/docs/twitter-api

[2] https://developers.google.com/youtube/v3/docs/comments/list

[3] https://en.wikipedia.org/wiki/Regular_expression

[4] https://www.nltk.org/api/nltk.tokenize.html

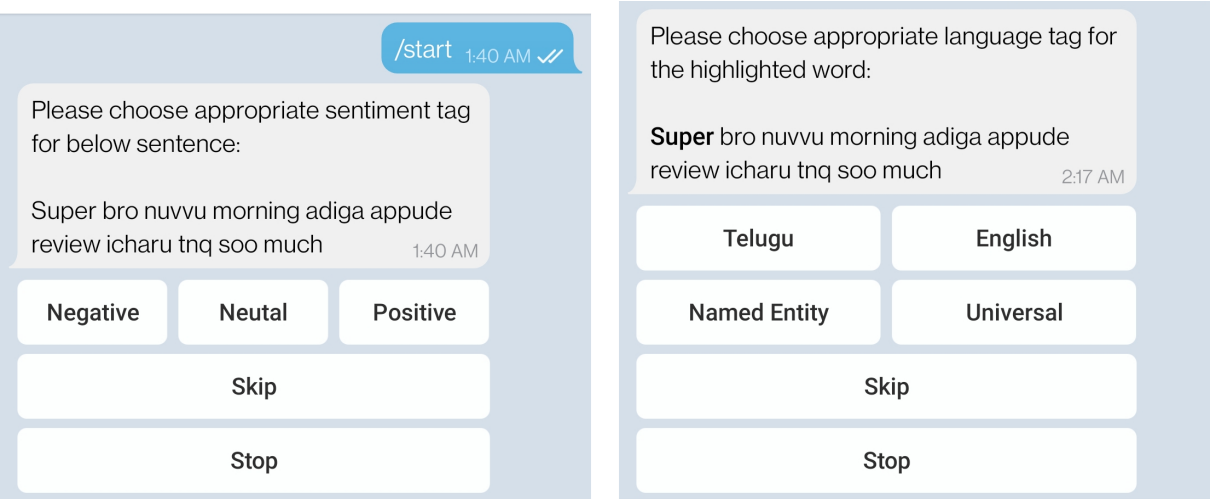

Figure 1: Screenshot of the Telegram Bot used for data annotation

After the word-level annotation, we removed the sentences having only English or only Telugu words.

The annotation was carried out by 5 Telugu native speakers who are also proficient in English. We developed an efficient annotation process with the help of Telegram Bot API[5], where the annotator can annotate from their Telegram App with a single tap on their device. Figure 1 shows a screenshot of the user interface of the Telegram Bot made for this task.

In most of the earlier works, the annotation task has been done by developing an application, which is usually web-based (Aprosio et al. (2020),(Wadden et al., 2020)). Using this process would either have device compatibility or user-experience issues. There will be a significant increase in application development overhead to address these issues. On the other hand, using a Telegram Bot API offered excellent user experience, lesser maintenance, lesser development overhead, and its availability in most devices (both mobile and desktop). Our annotators gave us positive feedback for the tool, pointing out the flexibility offered to perform the annotation task in any environment.

Inter Annotator Agreement: We calculated the Inter Annotator Agreement score using Cohen's Kappa score (Cohen, 1960) in order to assess the quality of the dataset. We have got a very high Cohen's kappa score of 92.3% for sentiment tags and 94.7% for language tags.

At the end of data annotation we got a total of 19,857 sentences of which 7,925 are Positive, 7,713 are negative and 4,219 are neutral sentences. The data is open-sourced[6] to encourage further research.

Sample Data:

Sentence: super/EN bro/EN nuvvu/TE morning/EN adiga/TE appude/TE review/EN icharu/TE tnq/EN soo/EN much/EN

Sentiment: Positive

4 Challenges in CMTET

The CMTET poses new challenges because of its unstructured nature due to the following phenomenons.

4.1 Informal Transliterations

Users in social media follow no standard when transliterating Telugu from Dravidian to Roman text. Hence many transliteration variants are observed in the CMTET. Below, we discuss the various transliterations observed.

4.1.1 Long Vowels

People tend to transliterate long vowels in many ways. For example, the word తిన్నావా (winnAvA) is transliterated into *tinnaavaa*

[5] https://core.telegram.org/bots/api

[6] https://github.com/ksubbu199/cmtet-sentiment

(repeating the long vowel twice) or just *tinnava* (not indicating the long vowel at all) or *tinnavaa*, *tinnaava* (mixture of double and single vowels).

4.1.2 Double Consonants

Similar to long vowels, even double consonants are transliterated in many ways. For example, తిన్నావా (winnAvA) is transliterated into *tinnava*, *tinava* or సరిగ్గా (sariggA) into *sariggaa*, *sarigaa*.

4.1.3 Aspirated Consonants

Aspirated Consonants are the syllables which require burst of breath to pronounce. In Telugu ఖ,ఛ,ఘ,ఠ,ఝ,భ,ధ are the aspirated consonants. In CMTET, we observed that these characters are transliterated in multiple ways. For example, ధర (xara) is transliterating into *dhara* or *dara*.

4.1.4 Homophones

Homophonic syllables are usually transliterated in multiple ways due to their nature of having same pronunciation with different spellings. For example, ఉన్నాయి (unnAyi): *unnayi*, *unnai* or ఎక్కడ (ekkada): *ekkada*, *aekkada*, *akkada*.

4.2 Informal Language

Some of the variations in the spellings are caused due to the lack of formal setting on social media. Following are some variations caused by the informal setting of social media.

4.2.1 Elongation

To express certain sentiments like excitement, users stretch some words in an informal setting. For example: *hellooo*, *niceeee*, *gooood*, *okayyy*, *bagundhiii*, *ichaavv*.

4.2.2 Shortening

Due to limited characters in Twitter, users tend to shorten words, yet capturing the word's phonetics. For example, *plz*, *grt*, *crct*, *ndku* (ఎందుకు:enxuku). In Telugu, shortening is usually done by dropping vowels or using single characters for double consonants.

4.3 Spelling and Typing Errors

Ritter et al. (2010), in their modeling of Twitter conversations, found that posts were "often highly ungrammatical, and filled with spelling errors" and resorted to selecting clusters of spelling variations manually. In CMTET, we observed spelling errors in both English and Telugu.

From the above, we can understand that CMTET has high entropy in spellings and thus poses a lot of challenges. To resolve these challenges, we propose an unsupervised data normalization technique for CMTET.

5 Data Normalization

In this section, we propose an unsupervised data normalization technique for normalizing CMTET. The architecture for data normalization can be found in Fig 2. As a first step, we performed elongation normalization and then used the language tags in the dataset to normalize Telugu words and English words separately. We then spell-checked English words using a defined similarity metric with an English dictionary. For Telugu words, we performed a two-stage normalization to normalize transliteration and spelling errors. The below sections explain this architecture in detail.

5.1 Elongation Normalization

To deal with the problem of Elongation (refer 4.2.1), we convert each character to lowercase, and then limit the repetitions of sequential characters to two. For example, *helloooo* to *helloo*, *gooood* to *good* and *bagundhiiii* to *bagundhii*. Errors persisting after this step like *helloo* and *bagundhii* are treated as spelling errors and are normalized in the next steps.

5.2 Normalizing English Words

To address spelling and typing errors, we have used dictionary-based spell-check with Levenshtein Distance (Levenshtein, 1966) as a similarity score between two words. We used SymSpell[7] to compute this efficiently.

5.3 Normalizing Telugu Words

In Telugu, the objective is to cluster the vocabulary into groups capturing the transliteration variants and spelling errors of each word. We propose a two-stage normalization for this. The below sections will explain this method in

[7] https://github.com/wolfgarbe/SymSpell

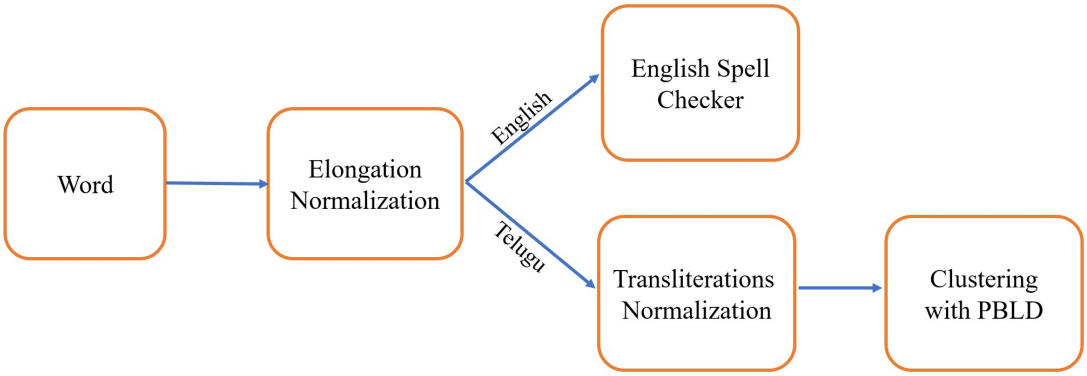

Figure 2: Proposed pipeline to normalize transliteration variants and spelling errors in CMTET

Standard Form	Meaning	Captured Variations
తరువాత (waruvAwa)	After	**tharuvatha**, taruvata, tharvatha, tarvataa, taruvatha
వీడికి (vEdiki)	For him	**veediki**, vediki, vedki, vidiki
చెత్త (cewwa)	The worst	**chethha**, chetha, chetha, cheta
అబద్ధాలు (abaxXAlu)	Lies	**abaddhaalu**, abbaddalu, abhadhalu, abadhal
ఎందుకు (enxuku)	Why	**endhuku**, endku, endhuku, enduk, endhuk, nduku
ఈరోజు (ErOju)	Today	**eerooju**, eroju eeroju, eorju, eorjuu

Table 1: Captured transliteration variations with the proposed normalization method

detail. Table 1 shows captured transliteration variations with this approach.

5.3.1 Normalizing Transliterations

In this stage, we aim to capture all the transliteration variations mentioned in Section 4.

- **Limiting Character Repetition to One:** This helps to address the issue of Long Vowels (refer 4.1.1) and Double Consonants (refer 4.1.2) transliterations. For example, *tinnaavaa* to *tinava* and *sarigga* to *sariga*.
- **Normalizing Aspirated Consonants:** The transliterations of aspirated consonants in Telugu i.e., *kh*(ఖ), *chh* (ఛ), *gh* (ఘ), *th* (ఠ), *jh* (ఝ) *dh* (ఢ) and *bh* (ఫ) are replaced with *k*, *ch*, *g*, *t*, *j*, *d* and *b* respectively. This will address the problem of Aspirated Consonant Transliterations (refer 4.1.3).

5.3.2 Clustering with PBLD

In this stage, we aim to normalize Homophones and spelling errors. As there is no standard dictionary for the transliterated Telugu text we aim to normalize the text by clustering them. We have experimented with Levenshtein Distance(LD)(Levenshtein, 1966) as a similarity score to cluster Telugu words. But, we observed a limitation that, LD treats all the characters equally leading to the clustering of wrong words. For example: According to LD, *rasthaaru* (రాస్తారు:rAswAru) and *vasthaaru* (వస్తారు:vaswAru) are unit distant. To address this issue, we propose a modified LD called Phonetic Based Levenshtein Distance (PBLD) with the following changes:

d	**Vocab. Reduction**		**Error %**	
	LD	**PBLD**	**LD**	**PBLD**
1	32.4%	5.5%	30.70%	47.69%
2	57.8%	10.17%	38.67%	69.38%
3	70.65%	18.12%	41.48%	83.01%

Table 2: Clustering error and Vocabulary Reduction with clustering transliteration variants with varying Edit Distance (***d***).

- Insertions and deletions are allowed only if they are vowels. This will address the issue of Shortening (refer 4.2.2) in Telugu.
- Characters can only be substituted with other characters if they have similar phonetics. This will address the variations in transliteration of Homophones (refer 4.1.4).

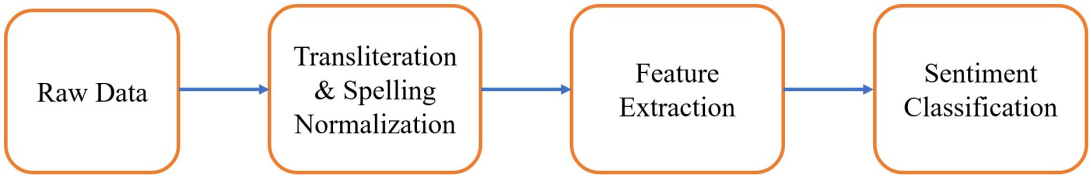

Figure 3: Proposed pipeline for Sentiment Analysis in CMTET

		Without Data Normalization			With Data Normalization		
Model	**Parameter**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1 Score**
NB	Overall	**77.33**	67.25	67.57	76.66	**67.48**	**67.80**
	Positive	67.05	**89.58**	**76.70**	**67.44**	88.73	76.64
	Negative	81.87	82.33	82.10	**81.98**	**83.33**	**82.65**
	Neutral	**83.08**	29.83	43.90	80.54	**30.39**	**44.13**
LR	Overall	74.73	74.27	74.47	**76.52**	**75.86**	**76.13**
	Positive	78.22	80.92	79.55	**79.65**	**83.44**	**81.50**
	Negative	82.53	82.46	82.50	**84.50**	**84.32**	**84.41**
	Neutral	63.44	59.43	61.37	**65.40**	**59.83**	**62.49**
RF	Overall	74.76	75.48	74.81	**75.81**	**75.81**	**75.67**
	Positive	**78.19**	79.11	78.65	77.49	**82.77**	**80.04**
	Negative	87.31	76.37	81.47	**88.43**	**78.92**	**83.41**
	Neutral	58.80	**70.96**	**64.31**	**61.52**	65.75	63.56
SVM	Overall	73.75	73.01	73.28	**74.98**	**73.61**	**74.05**
	Positive	78.55	76.80	77.66	**78.90**	**81.25**	**80.06**
	Negative	77.67	83.67	80.56	**79.78**	**84.71**	**82.17**
	Neutral	65.03	**58.56**	**61.63**	**66.00**	54.85	59.91
MLP	Overall	75.83	76.71	75.85	**78.08**	**78.8**	**78.31**
	Positive	**86.99**	76.17	81.22	83.98	**81.50**	**82.72**
	Negative	84.38	82.85	83.61	**88.37**	**84.02**	**86.14**
	Neutral	56.14	**71.11**	62.74	**61.89**	70.88	**66.08**

Figure 4: Quantitative results across various Machine learning models. It is observed that the metric values are lower without data normalization when compared to the proposed approach.

5.3.3 Error Analysis

In this section, we perform an error analysis on normalization of Telugu words with LD and PBLD. Clustering of words leads to the reduction of Telugu vocabulary in the dataset. We randomly picked clusters having a total of 500 words and observed a significant difference between the two methods in terms of clustering error. These metrics are reported in Table 2.

6 Method

In this section, we focus on explaining our Sentiment analysis pipeline. Fig. 3 shows our end to end approach wherein we take raw data i.e. a sentence and perform data normalization (transliteration and spelling normalization) as explained in section 5. Once we have this normalized data we perform feature extraction using N-Grams and Term-Frequency and Inverse-Document Frequency (TF-IDF) (Chowdhury, 2010). These features are then passed to our sentiment classification models which outputs one of the labels namely positive, negative and neutral.

We experimented on Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF) and Multi Layer Perceptron (MLP) classification models to classify the final sentiment.

7 Experiments and Results

The motivation of the current work is achieve better Sentiment Analysis in CMTET using machine learning models. Majority of the existing approaches majorly focus on single language sentiment analysis (Zulkifli and Lee,

Text	Expected	Predicted
vadu manchi director em kaadu, oka hit kuda leduu *(He is not a good director, he don't even have a successful movie)*	Negative	Positive
feel ayithae cheppu bro, hurt cheyali ani analedhu *(Please let me know if you got hurt, I didn't mean to hurt you)*	Neutral	Negative
climax maataram shawshank redemption la undhi bro <3 *(Bro, climax is like Shashawnk Redemption <3)*	Positive	Neutral

Table 3: Error Analysis with MLP

Models	Without Data Normalization	With Data Normalization
NB	74.06%	74.23%
LR	76.95%	78.76%
RF	76.31%	77.65%
SVM	75.59%	76.98%
MLP	77.69%	80.22%

Table 4: Accuracy metrics of various ML models with two different approaches

2019), (A. Al Shamsi et al., 2021), (Mukku et al., 2016) and rarely consider Code-mixed text (hin). The central focus in the experiments is to extensively benchmark across standard machine learning models on CMTET.

Table 4 compares our novelty in data normalization process to that of a naive one across various Machine learning models. We put forward an extensive set of quantitative metrics comparing precision, recall and F1-Score across classes (positive, negative and neutral). From table 4, it can be seen that by employing our Data Normalization technique, there is an increase in the Overall model's F1 score and accuracy. From table 4, in NB, LR, RF, SVM and MLP we can see an increase in total accuracy of 0.17%, 1.81%, 1.34%, 1.39% and 2.53% respectively. Also, the designed MLP architecture outperforms the other models by more than 1.5%.

8 Error Analysis and Observations

The task of Sentiment Analysis on social media text is difficult in itself as it often contain sarcastic text. The other major challenge is the use of emoticons and emojis in social media text which often drive the sentiment of the sentence. Table 3 shows some of the examples that are incorrectly predicted by our best model. In some cases, overall sentiment of the sentence mightn't be just based on the positive or negative words used, but on the context in which these words are used. This can be observed in first and second examples shown in Table 3. In the third example, though there are no direct positive words, the sentiment of this sentence is because of the positive sentiment associated with *The Shawshank Redemption*[8] and the emoticon used.

9 Conclusion and Future Work

In this paper, we published a huge annotated dataset for sentiment analysis in CMTET to encourage further research. Also, we presented a pipeline for this task with a novel data normalization technique. For each model, we have shown quantitatively that the proposed data normalization improves the overall performance across various metrics (precision, recall and F1-score) (refer Table 4). To the best of our knowledge this is the first such method carrying out sentiment analysis on CMTET.

Although the current work addresses some of the most crucial challenges in sentiment analysis on CMTET it can further be extended to other languages. The proposed data normalization technique can also be leveraged in various other NLP tasks. We can further make the bench-marking more extensive by including other deep learning models like LSTM, RNNs and CNNs.

References

Arwa A. Al Shamsi, Reem Bayari, and Said Salloum. 2021. Sentiment analysis in english texts. *Advances in Science Technology and Engineering Systems Journal*, 5:1683–1689.

[8] https://www.imdb.com/title/tt0111161/

Alessio Palmero Aprosio, Stefano Menini, and Sara Tonelli. 2020. Creating a multimodal dataset of images and text to study abusive language.

Monika Arora and Vineet Kansal. 2019. Character level embedding with deep convolutional neural network for text normalization of unstructured data for twitter sentiment analysis. *Social Network Analysis and Mining*, 9.

Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani. 2019. Normalization of Indonesian-English code-mixed Twitter data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 417–424, Hong Kong, China. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020. A sentiment analysis dataset for code-mixed Malayalam-English. 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 177–184, Marseille, France. European Language Resources association.

Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. 2018. Sentiment analysis of code-mixed languages leveraging resource rich languages.

G. Chowdhury. 2010. *Introduction to Modern Information Retrieval, Third Edition*, 3rd edition. Facet Publishing.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Geetika Gautam and Divakar Yadav. 2014. Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In *2014 Seventh International Conference on Contemporary Computing (IC3)*, pages 437–442.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards subword level compositions for sentiment analysis of Hindi-English code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491, Osaka, Japan. The COLING 2016 Organizing Committee.

A Kalaivani and D Thenmozhi. 2020. Ssn_nlp_mlrg@ dravidian-codemix-fire2020: Sentiment code-mixed text classification in tamil and malayalam using ulmfit. In *FIRE (Working Notes)*, pages 528–534.

Moshe Koppel and Jonathan Schler. 2006. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22:100–109.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models.

Pruthwik Mishra, Prathyusha Danda, and Pranav Dhakras. 2018. Code-mixed sentiment analysis using machine learning and neural network approaches.

Sandeep Mukku, Nurendra Choudhary, and Radhika Mamidi. 2016. Enhanced sentiment classification of telugu text using ml techniques.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, Los Angeles, California. Association for Computational Linguistics.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview of the fire 2013 track on transliterated search. In *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, pages 1–7.

Nazanin Sabri, Ali Edalat, and Behnam Bahrak. 2021. Sentiment analysis of persian-english code-mixed texts.

Shashank Sharma, PYKL Srinivas, and Rakesh Chandra Balabantaray. 2015. Text normalization of code mix and sentiment analysis. In *2015 international conference on advances in computing, communications and informatics (ICACCI)*, pages 1468–1473. IEEE.

Gurpreet Singh Josan Sukhpreet Kaur. 2021. Sentiment analysis of code-mixed language. *International Journal of Advanced Science and Technology*, 30(01):01 – 11.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims.

Siddharth Yadav and Tanmoy Chakraborty. 2020. Unsupervised sentiment analysis for code-mixed data.

Nor Saradatul Akmar Zulkifli and Allen Wei Kiat Lee. 2019. Sentiment analysis in social media based on english language multilingual processing using three different analysis techniques. In *Soft Computing in Data Science*, pages 375–385, Singapore. Springer Singapore.