



Low resource language specific pre-processing and features for sentiment analysis task

Loitongbam Sanayai Meetei^{1,2}  · Thoudam Doren Singh^{1,2} · Samir Kumar Borgohain² · Sivaji Bandyopadhyay^{1,2}

Accepted: 6 May 2021 / Published online: 2 June 2021
© The Author(s), under exclusive licence to Springer Nature B.V. 2021

Abstract Sentiment analysis is a classification task where polarity of textual data is identified, i.e. to analyze whether a sentence or document expresses a negative, positive or neutral sentiment. Manipuri is a less privileged, highly agglutinative and tonal language. Despite being a scheduled language of Indian Constitution, it is also a resource constrained language. In this work, we report the sentiment analysis for Manipuri using different types of machine learning based approaches. The dataset used in our work is collected from local daily newspaper. The novelty of this work is that we carry out language specific pre-processing tasks such as transliteration, building negative morpheme based lexicon and filtering of noisy words. Using them as additional linguistic features in our models improves the classification result in terms of precision, recall and F-score. The ensemble voting of best three classifiers based on TF-IDF perform better than BM25 based classifiers and other stand-alone classifiers. Based on this result, we attempt to classify the sentiment of news articles during a certain period of time. Further, we report the finding of deep learning based approaches on the same dataset.

Keywords Low resource · Pre-processing · Morphology · Sentiment analysis · Machine learning · Deep learning · Ensembled classifier · TF-IDF · BM25 · Manipuri

✉ Loitongbam Sanayai Meetei
loisanayai@gmail.com

Thoudam Doren Singh
thoudam.doren@gmail.com

¹ Center for Natural Language Processing (CNLP), National Institute of Technology Silchar, Silchar, Assam, India

² Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam, India

1 Introduction

Sentiment analysis can be defined as the process of classifying an opinionated textual data into its polarity i.e as a negative, positive or neutral opinion. Tremendous growth in digital data, especially through social networks and blogs has fueled the interest in sentiment analysis because of its usefulness. For instance, a manufacturing company marketing their products can use sentiment analysis for improving their product's qualities and services by analyzing the products' reviews (Zhang et al., 2012). Sentiment analysis can also be helpful in minimizing the crisis level in any state or country by constantly monitoring current updates in social media conversations (Johansson et al., 2012).

In textual data, sentiment analysis has been carried out at the following levels:

- Document level (Pang et al., 2002), where polarity of the whole document is determined. The document can consist of single or multiple sentences.
- Sentence (Yu & Hatzivassiloglou, 2003) and phrase level (Wilson et al., 2005), typically consists of two steps (Khan & Baharudin, 2011). First, a sentence is checked whether it is subjective or not. Then, filtered subjective sentences are classified into positive or negative sentences.
- Word level (Kim & Hovy, 2004), where polarities towards different entities are determined.

Sentiment analysis at sentence or phrase level can be helpful in natural language processing applications such as mining product reviews, opinion oriented information extraction (Nasukawa & Yi, 2003), etc.

Further, data for sentiment analysis task can be collected from online discussion forums, articles from newspapers, online reviews, social media, etc. But first the data collected from such platforms are to be cleaned and standardized through a pre-processing step as the raw user generated data are usually unstructured and not analyzable for carrying out sentiment analysis. Stop words removal (Denecke, 2008), filtering of URL links and special symbols (Pak & Paroubek, 2010), polarity annotation (Wilson et al., 2005), etc. are some of the pre-processing tasks. In case of dataset collected from social media sites, special symbols such as emoticons, exclamatory marks, etc. can be used as a feature in sentiment analysis (Mishne, 2005). The pre-processed data is transformed into a feature where weights are assigned to each feature, which helps in selecting maximum contributing attributes of the dataset. These features are then used to classify the dataset into two or more classes using different types of classification methods.

Sentiment classification methods can be grouped into three different types namely, supervised (Pang et al., 2002), semi-supervised (Goldberg & Zhu, 2006) and unsupervised (Hu et al., 2013). Most of the work on sentiment classification is carried out using supervised methods by employing various machine learning techniques. In supervised machine learning technique, the lack of an organized polarity oriented dataset is one of the challenges.

The main objective of our work is to perform sentiment analysis for Manipuri language where orientation of the text is classified into either negative, positive or neutral sentiment. In brief, Manipuri is the lingua franca of Manipur, a northeastern state of India. It is not only the official language of Manipur but also included in the 8th Scheduled of Indian Constitution. Its usage can also be found in other neighboring states like Assam and Tripura. Also outside of India, Manipuri is spoken in Burma and Bangladesh. Manipuri is a Tibeto-Burman language, a sub-family of Sino-Tibetan languages, which has its own script, with the number of speakers approximately around 3 million. Bengali script was used to represent Manipuri in all academic textbooks and literary works instead of the indigenous Meetei Mayek script till the year 2006. In 2006, the state government reintroduced the indigenous script and it was adopted in all the schools and educational institutions in Manipur starting at the primary level, thereby upgrading yearly. Because of the above changes, there is a large gap between two generations: one who was educated in Bengali script Manipuri and the other educated in Meetei Mayek script Manipuri. Though Manipuri is still being written using both Bengali and Meetei or Meitei Mayek script (hereon Meetei Mayek script), now the new generations are leaning more towards the usage of Meetei Mayek script.

Now coming back to the objective, very limited availabilities of language tools such as part of speech (POS) tagger for Manipuri restricts us from exploring different methods of sentiment analysis. Also, research work on sentiment analysis for Manipuri language using machine learning technique is a very recent activity. In this paper, we applied different machine learning techniques to perform classification of sentiment analysis at the sentence level for Manipuri language. The dataset is collected from news article of local daily newspapers. The work could be helpful in assisting Non-Governmental Organizations (NGOs) or certain government agencies in detailed analyzing and understanding the sentiments of a community for any political or non-political developments.

The rest of the paper is structured as follows: Sect. 2 reviews previous relevant research. Section 3 presents our approach and system set up. Section 4 details the performance evaluation of our findings. Finally, Sect. 5 summarizes the conclusion of our work and presents avenues for further research.

2 Related works

In the field of research, whether in industry or academic, English text has been the main focus because of its vast usage and availability in digital form. As per the literature review, previous studies of sentiment analysis have almost exclusively focused on well-studied languages such as English, German, Chinese, etc. Over time, many researchers have started sentiment analysis task in low resource languages (El-Haj et al., 2015; Le et al., 2016; Gangula & Mamidi, 2018). Early studies on sentiment analysis include (Pang et al., 2002). The author reported that the performance of sentiment classification using machine learning techniques is better than human-produced baselines. The author uses three machine learning techniques, namely Naïve Bayes, Support Vector Machine (SVM), and Maximum

Entropy on a dataset collected from movie reviews. Using a 3-fold cross-validation, a comparison was made on feature presence and feature frequency. The presence of a feature represents the occurrence of a feature i.e. boolean frequency while feature frequency represents the frequency of occurrence of a feature i.e. term frequency. Feature presence was reported to perform better than feature frequency and among the three classifiers SVM showed the best result.

Na et al. (2004) reported work on an automatic sentiment classification using a product review dataset. The work focus on the advantage of using a machine learning classifier with different text features such as a single word, words belonging to specific categories, and POS-tagged words were used. Text data were characterized using different terms weighting schemes such as binary, term frequency, and TF-IDF (Term Frequency-Inverse Document Frequency). The classification was carried out using SVM classifier. A dataset comprising of words belonging to certain categories was reported to outperform other methods. An improvement in the classification result was observed in the study conducted on negation phrases through a linguistic approach.

Different types of text representation have been carried out for the feature characterization in the field of sentiment analysis. In the work of Sixto et al. (2016), the authors experimented with Okapi BM25 ranking function to evaluate its performance in the sentiment analysis task. Using various language models, the sentiment analysis was carried out on the tweets in Spanish language. With SVM as the base classifier, the classification was also performed with other machine learning classifiers such as Logistic Regression, Gradient Boosting, and a VoteClassifier (Sixto et al., 2016). The author reported that the optimal values of the free parameter used in Okapi BM25 for sentiment analysis are different from the typical values used in other research fields.

The application of sentiment analysis has also found its way into the medical field. One such work is reported by Niu et al. (2005). Using dataset collected from medical data, a supervised machine learning technique along with various language models was employed to carry out a sentiment analysis at the sentence level. Instead of determining personal opinions which are a typical tasks of sentiment analysis, the work focused on polarity information on medical outcomes of patient's records. The dataset is classified into four classes using SVM. Apart from the language model, linguistic features and domain knowledge were incorporated as part of the features during the classification process. The author reported an improvement in the result when linguistic and domain-related features were added.

Albayati et al. reported work on sentiment analysis for Korean, a morphologically rich language. The authors experimented with the practicability of linguistic approach by applying a new morphological chunking method. The approach also focuses on the role of contextual shifters (Jang & Shin, 2010). Part of Speech (POS) tagging was carried out using an open source probability-based Korean morphological analyzer (KTS). The dataset collected from news and movie reviews were manually annotated by two native Korean annotators. With the application of 5-fold cross-validation, the classification was conducted using TF-IDF as the term weighting scheme and SVMlight as the classifier. The authors concluded that the use of language-specific features was effective in the field of sentiment analysis.

With the advent of deep learning, Long Short-Term Memory Network (LSTM) (Hochreiter & Schmidhuber, 1997) and Convolutional Neural Networks(CNN) are widely used for text classification. Dashtipour et al. (2020) proposed a concept-level sentiment analysis for the Persian language by incorporating linguistic rules and deep learning to improve polarity detection. Addressing the current Persian sentiment analysis approaches which are based on the frequency of word co-occurrence, where word order and hierarchical relation between them are not considered, the model focus on the dependency relations between keywords, word order and, individual word polarities. Albayati et al. reported work on sentiment analysis for Arabic, another language with a rich morphology structure, by employing deep learning models. Instead of hand-crafted features, LSTM neural networks with the word embedding layer is used in the experiment.

With social media becoming a source to harvest various kinds of data, resources for natural language processing tasks can be obtained from social media (Sixto et al., 2016; Singh et al., 2021). Analysis of consumer reviews will determine the potential goods and services of a company. Cambria et al. (2017) stated that affective computing and sentiment analysis may boost the ability of customer engagement to discover what features customers enjoy. However, such user-generated content is often made up of multilingual text. Vilares et al. (2018) propose a reproducible method to generate SenticNet (Cambria & Hussain, 2015) for a variety of languages resulting in BabelSenticNet, a concept level knowledge base for multilingual sentiment analysis. Lo et al. (2017) present a detailed review on works related to multilingualism. The authors identified subjectivity and polarity detection as the two main approaches in sentiment analysis. The authors proposed creating a bilingual dictionary and subjective lexicon as the methods for generating lexicons for languages with limited resources. Unlike the texts of other social media, texts in Twitter is limited to 140 characters. Using such micro text, Sixto et al. (2018) investigates the structured information of social networks for subjectivity detection.

In the context of the Indian language scenario, sentiment analysis for three Indian languages namely, Bengali, Hindi, and Telugu was reported by Das and Bandyopadhyay (2010). The authors developed SentiWordNet for each of the languages separately using various approaches such as bilingual dictionary-based, WordNet-based, and corpus-based. A polarity classifier developed using the SentiWordNet along with linguistic features, manual validation, and online games were used for evaluating the polarity score (positive and negative). The online game based evaluation approach where a user can tag the polarity of the text data could be of use in creating linguistic data. Works on sentiment analysis on Manipuri are carried out by Nongmeikapam et al. (2014) and Singh et al. (2021). Nongmeikapam et al. (2014) studied the sentiment analysis for Manipuri language at the document level. Dataset collected from letters to the editor of local newspapers were used for the experiment. Conditional Random Field (CRF) based Part of Speech (POS)tagger was used to identify the verbs. After identifying the polarity of the verbs, the highest count of polarity in the document was used to determine a sentiment as positive, negative, or neutral. Using social media comments in Manipuri, Singh et al. (2021) report the findings on sentiment analysis with lexicon based methodologies,

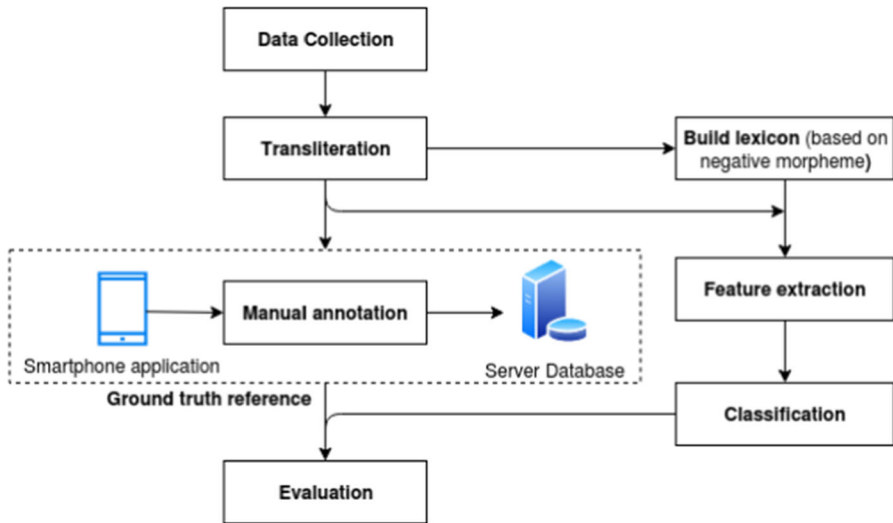


Fig. 1 Framework of our model

traditional machine learning and deep learning techniques. The authors also report the significance of pre-processing and feature engineering on these social media comments for the sentiment analysis.

3 Architecture

A pictorial representation of the framework for sentiment analysis for the Manipuri language is shown in Fig. 1. The architecture is grouped into four stages: data collection, pre-processing, feature extraction, and classification.

The collection of textual data in the Manipuri language is carried out in the first stage of the model. The second stage, i.e. pre-processing aims at standardizing the dataset. It consists of transliteration, manual annotation, and tokenization. In the third stage, TF-IDF and Okapi BM25 are used to characterize the processed textual data separately along with an extracted list of linguistic features into feature vectors. Finally, sentiment analysis is carried out by applying machine learning classifiers on the feature vectors. The details of the stages are elaborated in the following subsections.

3.1 Data collection

As of today, no significant digital resource is available to carry out sentiment analysis of Manipuri. The dataset used in our experiment is collected from the local daily newspapers (Hueiyen Lanpao, *HL*¹ and The Sangai Express, *SE*²) based in

¹ <http://hueiyenlanpao.com/>.

² <http://thesangaexpress.com/manipuri/>.

Manipur in the form of portable document format (pdf). Unlike the informal online comments or social media data, our news articles are formal one. The collected dataset comprises of 120655 sentences. The dataset is grouped into two categories: 5009 labeled sentences (hereon $D1^3$) which is our gold standard and 115,646 unlabeled sentences (hereon $D2$) which will be used to classify the sentiment of news articles. Our gold standard dataset $D1$ is collected from HL and is written in Meetei Mayek script. The data set $D2$ in Bengali script is collected from the corpus reported by Singh and Bandyopadhyay (2010). A common representation of the news corpus required for the task is explained in the pre-processing step.

3.2 Pre-processing

Most of the online news texts are unstructured. So, pre-processing of these texts is a crucial step before carrying out any task. Improvement in the results of sentiment analysis was reported in Haddi et al. (2013) and Jianqiang and Xiaolin (2017) where the authors carried out various pre-processing methods on social media texts in English. Some of the pre-processing methods include white space removal, stemming, removal of stop words, removal of numbers, removal of URL links, negation handling, replacing negative mentions, reverting words that contain repeated letters in their original form etc. While some of the pre-processing methods mentioned are language-independent others are applicable only for language-specific and social media text. In this work, we focus on the language-specific pre-processing tasks such as transliteration, building negative morpheme-based lexicon and filtering of noisy words. Early work of a bidirectional transliteration system of Meetei Mayek and Bengali script is reported by Singh (2012). Nowadays, the native speakers of this language use the Roman script instead of the Meetei Mayek script or the Bengali script for short text messages. Some of the possible reasons being the generation gap and the popularity or handy to type Roman script from the devices for the social media platform. In order to bring a common platform, we have transliterated our dataset which is in the Meetei Mayek and Bengali script to the Roman script. We carry out the pre-processing step of the two dataset ($D1$ and $D2$) separately.

3.2.1 Pre-processing of dataset $D1$

The pre-processing of the dataset $D1$ is carried out as follows:

1. Transliteration of Meetei Mayek script to the Roman script is carried out first.
2. This is followed by the manual polarity annotation of the transliterated dataset which is considered as a gold standard dataset.

Letters in the Meetei Mayek script⁴ are grouped as eeyek eepee, lom eeyek, lonsum eeyek, cheitap eeyek, cheising eeyek and khutam eeyek. The news articles data in

³ <https://github.com/LSMeetei/sentiment>.

⁴ <http://tabish.freeshell.org/eeyek/script/>.

Meetei Mayek script are transliterated into Roman script using a transliteration module developed in-house. The transliteration process is divided into two steps as follows:

1. Map the characters from newspaper font to Unicode characters of Meetei Mayek script. By the time of collecting the text, the font used in the newspaper is RATHA TrueType font which is available from the website.⁵ Here, a total of 54 Meetei Mayek characters present in our dataset are mapped to their corresponding code points (range: $U+ABC0$ to $U+ABFF$) in the Unicode block for Meetei Mayek script. The character with code point $U+ABD3$ is not present in our dataset and is left empty.
2. Map the characters obtained in step 1 to the corresponding transliterated Roman character(s). In this step, the 53 characters were grouped into 5 lists as follows:
 - Numerical: CHNUM. e.g: ᠠᠨᠠ ᠠᠨᠠ
 - Vowels: CHV. e.g: ᠠ
 - Lonsum consonants: CHL. e.g: ᠠᠨ
 - Consonants which are not Lonsum: CHNL. e.g: ᠠᠨᠠ
 - Joining line character: CHL: ᠠᠨᠠ

In the transliteration step, the whole dataset is split into word level. Each of the words is split into a list of characters. On mapping the character level Meetei Mayek script to its corresponding Roman character(s), we found the presence of extra vowel *a* which leads to a loss of actual pronunciation of the word. This is minimized by deletion of a schwa. A schwa is a mid-central vowel sound in English denoted by the symbol ə, which in our case is the vowel *a*. Algorithm 1 summarizes the rules followed for the schwa deletion.

Algorithm 1: Transliteration of Meetei Mayek script to Roman script.

1. For any three adjacent characters XYZ, if Z is a joining line then check
 - if $X \notin \text{CHL}$, then delete the extra “a” of X.
 - if $Y \notin \text{CHL}$, then delete the extra “a” of Y.
 2. For any two adjacent characters XY, if $Y \in \text{CHV}$ and $X \notin \text{CHL}$, then delete the extra “a” of X.
-

A sample of our transliterated input and output is shown in Table 1.

The transliterated sentences of dataset *DI* are further subjected for manual annotation in a distributed fashion by native speakers. For this purpose, we upload the dataset on our web server. The dataset is prepared for annotation through an android application developed in-house. The application was distributed to three native Manipuri speakers by assigning separate user credentials for each of them. The polarity annotation by them through the application is recorded back in the

⁵ <http://www.e-pao.net/>.

id	topic_id	text	positive	negative	neutral
1	1	mioiba eikhoina nong hek nganbadagi numit hek taba...	3	0	0
2	1	chahi nouriba angangsingna yamna pamna chanariba p...	2	0	1
3	1	masidagi hendokladuna emung amana sijinnariba chan...	2	0	1
4	1	pubada nungngaiba, masada paksindaba, mamal hongba...	1	2	0
5	1	nongnganbadagi numit taba phaubada yum amada pusil...	2	0	1
6	1	lambida lamjauda, loubuk lammangda, lukhak komda n...	2	1	0
7	1	asumna eikhoi meoibana sejinnaba yadraba potlamsin...	1	1	1

Fig. 3 Sample from the server

Figure 2a shows the list of article heading along with the number of sentences. When the user clicked on any of the headings, a list of sentences under that heading is shown to the user one after another (Fig. 2b). As shown in Fig. 2b, the user can opt the sentiment of the sentence as negative, positive or neutral. After selecting the sentiment, the user can proceed to the next sentence until all the opinion of the sentences under the selected heading is captured. The selected opinions are then submitted back to the repository. Figure 3 shows the count of opinion annotated as recorded in the server along with the listing of sentences. In case, the opinions of each of the annotators are different in all the cases, a voting system is employed to decide the polarities of each sentence. The sentence with a score of two or more in the positive, negative or neutral is considered to be a positive, negative and neutral sentence respectively. The sentence with a score of one each for positive, negative and neutral is taken as a neutral sentence. After the voting system is employed, there are 3176 positives, 377 negatives and 1456 neutral sentences from the collection of 5009 sentences.

Manipuri is a tonal, morphologically rich and highly inflectional language. Most of the negative sense of a text can be identified using suffix. As described in the work of Manipuri morphological analyzer by Singh and Bandyopadhyay (2006) in the role of affixes on a sentence level, negative sentences are formed in Manipuri by the suffixation of negative markers, namely, $\sim te/ \sim de, \sim loy/ \sim roy$ to the verb. Some of them are negative words with these markers (e.g. *natte* meaning *not*). Combining these markers along with some other markers, we extracted a list of words with the following suffix: $\sim te, \sim de, \sim loi, \sim roi$ (allomorph of $\sim loy$ and $\sim roy$ respectively), $\sim dre, \sim tre, \sim dri, \sim tri$. The dataset *D1* consists of 326 unique words (hereon NM_1) with the above negative markers as its suffix. Sentences that contain any of the NM_1 words are identified from the 3176 positive and 1456 neutral sentences. A total of 907 sentences were extracted. There are 589 from positive dataset and 318 from neutral dataset. These 907 sentences were found to express the negative sentiment after further examination. After these new findings, *D1* is finally re-grouped into 3176 positive, 1284 negative and 1138 neutral sentences as shown in Fig. 4. The statistics of dataset *D1* is shown in Table 2.

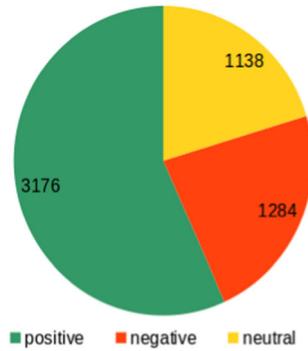


Fig. 4 Dataset D1

Table 2 Statistics of dataset D1

Number of sentences	Number of tokens	Average sentence length
5009	85969	17.16

Table 3 A sample of transliteration from Bengali script to Roman script

Text in Bengali script	Transliteration text in Roman script
<p>চিংনিংগাই ওইবা মঙংনা অঙাং অনী সকুলদপী মাঙলুবদপী মীয়মনা ইরাং বৌরকতুনা সকুল শঙ মৈ থাংগে তৌরমেল ইরাংত্ৰ পুলিসনা লাকশিনবিবগা লোয়ননা অঙাং অনীহুসু লৌখতলুৰে অঙাংশিং অহুদি খুৰাই হৈক্ৰুম- খোঙনা লৈবা লৈমরাম বিৰবাহপী মচানুপা ৰুদৰাজিত, 12 অমসুং মফম অহুৰাইদা লৈবা যাত্ৰা ইংলিস সকুলপী টোকিদাৰ ওইবা জোনি থাপাগী মচানুপা অকাশ থাপা, 12 নি মখোয় অনীমক সকুলদুদা ক্লাস ফাইবতা তম্মিননবনি মখোয় অনী গুসি নুংথিন পুং 1.10 রোম তাবনা সকুলপী হাল্ফ থাৰবা মতুদা মাঙবিবনি </p>	<p>chingnaningnai oiba maongda gangang ani skuldagi manglubadagi miyamna erang houraktuna skul shang mei thage touramle. erangdu pulisna lakshinkhibaga loinana gangang anidusu loukhature. gangangshing adudi khurai heikruma-khongda leiba leimaram birabahugi machanupa rudrajit, 12 amasung mapham aduwaida leiba yatra englis skulgi choukidar oiba joni thapagi machanupa akash thapa, 12 ni. makhoi animak skulduda klas phaibata tamminnabani. makhoi ani ngasi nungthin pung 1.10 rom tabada skulgi halph tharaba matungda mangakhibani.</p>

Translation in English: A mob tried to burn down a school as two kids went missing from the school in a suspicious manner. The police controlled the mob, as well as the two kids, were recovered. One of the two kids is Rudrajit (12), son of Laimaram Birbahu of Khurai Heikru makhong and the second kid is Akash Thapa (12), son of Joni Thapa who is working as the gatekeeper in Yatra English school of a nearby locality. Both the kids were studying in Class 5. Both of them were missing from the school after half time around 1:10 pm

3.2.2 Pre-processing of dataset D2

As for the dataset D2, the textual data in Bengali script were again transliterated into Roman script using transliteration module developed in-house. In the transliteration process, we collected a total of 59 unique characters consisting of vowels, and consonants. A simple rule of character to character(s) mapping is followed where all

the 60 characters have been mapped to their corresponding transliterated roman character(s). Using the above 59 characters, a list of 4 subsets are constructed that contains:

1. Vowels: CHV. e.g: অ ি
2. All consonants: CHC. e.g: ক খ
3. Consonants which are mapped to one Roman character: CH1. e.g: ক
4. Consonants which are mapped to two Roman characters: CH2. e.g: খ

A similar approach of character mapping of Meetei Mayek script to Roman character(s) is also carried out in the case of transliteration of Bengali script to the Roman script. During the mapping, we found the presence of a cluster of consonants which leads to a loss of actual pronunciation of the word. The issue is minimized by using the addition of a schwa ə, which in our case is the vowel *a*. Algorithm 2 summarizes the rules followed for the schwa addition.

Algorithm 2: Transliteration of Bengali script to Roman script

1. For any two adjacent characters XY, where X and Y belongs to CHC, add “*a*” between X and Y.
 2. If the last character of a word belongs to CH2, then add “*a*” at the end.
-

A sample of the transliterated text of Bengali script to Roman script is shown in Table 3.

After the transliteration of the dataset *D2*, we have also extracted the list of words with the following suffix: $\sim te$, $\sim de$, $\sim loi$, $\sim roi$, $\sim dre$, $\sim tre$, $\sim dri$, $\sim tri$. The dataset *D2* consists of 2660 unique words (hereon NM_2) with the above negative markers as its suffix.

Despite most of the Manipuri words being transliterated correctly to Roman script, the transliteration module could not correctly transliterate all the words which are in the Meetei Mayek script and the Bengali script. The phoneme to lexeme of Manipuri has certain orthographic variations as the texts are written as they are pronounced. In some of the Manipuri words, extra *a* is found in the output after transliteration. The transliteration tasks implemented in Java for Meetei Mayek script to Roman script and Bengali script to the Roman script are carried out in the normalized format, i.e. lower case Roman characters.

3.2.3 Filtering of noisy words from dataset *D1* and *D2*

For further processing, each of the transliterated sentences in dataset *D1* and *D2* are tokenized to check the presence of alphanumeric words, numerical character(s) or special character(s). If any matches are found, the word or character is removed from the sentence. In the orthography of Manipuri, there are few words with two characters of Meetei Mayek and one character in the corresponding Bengali script. On transliterating such words into Roman script results to a word of two characters. For example: ᱠᱟ in Meetei Mayek corresponds to ঐ in Bengali script and its

transliteration in Roman script (TRC) is “*ei*”, meaning “*I*” or “*me*” in English. Such words are hardly found in the news articles and its presence is in inflected form such as ᱫᱷᱟ (TRC: “*eina*”). As such, almost all the words that appear in the news articles whose length is less than or equal to two tends to be a foreign word or an abbreviation letter. These kind of words are filtered from our dataset before implementing the following steps of the feature extraction process.

3.3 Feature extraction

In order to feed the textual data to machine learning classifiers, it is converted to features characterized by numeric values. It can be in the form of binary, i.e. presence or absence of a term in the sentence, or into the frequency of occurrence of a term. The above two conversion methods may favour one frequent term over the more distinct term or treating every term equally. In order to avoid such cases, the text data is transformed using TF-IDF term weighting scheme or Okapi BM25 ranking function to reward rare terms.

3.3.1 TF-IDF

TF-IDF is a method that assigns a weight to a term in a sentence which helps in determining the significance of the term. TF-IDF comprises of two parts, namely *tf* (term frequency) and *idf* (inverse document frequency). *tf* calculates the frequency of term *t* in a sentence *s*. The higher the frequency of a term, the importance of the term in the sentence is more. It is represented as:

$$tf(t, s) = f_{t,s}. \quad (1)$$

idf is the inverse of *tf* where lower the frequency of a term, higher is the *idf* coefficient of the term in a sentence. It is calculated as:

$$idf(t) = \log \frac{n}{S(t)}, \quad (2)$$

where *n* is the number of sentences in the dataset and *S(t)* is the number of sentences where the term *t* occurs.

By multiplying the above Eq. (1) and Equation (2), we get TF-IDF:

$$tf - idf = tf(t, s) \cdot idf(t). \quad (3)$$

3.3.2 Okapi BM25

Okapi BM25 is also a similar type of weighting scheme as TF-IDF but with a ranking function. Developed by Robertson et al. (1995), Okapi BM25 is used as a ranking function to compute the relevance score of a document with respect to a query. In our case, we use to compute the score of each term in a sentence with respect to the dataset. The BM25 score of a term *t* of a sentence *S* in a dataset *D* is computed as follows:

$$BM25(t) = IDF(t) \frac{f(t) \cdot (k + 1)}{f(t) + k \cdot (1 - b + b \cdot \frac{|D|}{avgSL})}, \quad (4)$$

where, $f(t)$ is the frequency of the term t in the dataset D . $|D|$ is the total number of terms in the dataset D . $avgSL$ is the average sentence length in the dataset. k and b are free parameters which are set to 2.0 and 0.75 respectively.

$IDF(t)$ is calculated as:

$$IDF(t) = \log \frac{N - n(t) + 0.5}{n(t) + 0.5}, \quad (5)$$

where N is the total number of sentences in the dataset D and $n(t)$ is the number of sentences containing the term t .

Feature extraction helps in excluding less informative and non-redundant data. The feature selection is performed to identify significant features to enhance target result. There are many methods for feature selection, one being the removal of features with low variance by using a custom threshold. Terms related to language-specific can also be put to use as a feature in the feature selection. We use the negative markers NM_1 and NM_2 as language-specific features with higher weightage. In our experiment, implementation of the above process is carried out using scikit-learn⁶ Pedregosa et al. (2011) package and Okapi BM25⁷ in Python along with other libraries such as numpy⁸ etc.

3.4 Classification using machine learning classifiers

Feature values obtained from the transformation of text data into numerical values are used for different classification methods. Several experiments are performed with the following classifiers: Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Gaussian Naive Bayes (GNB), Decision Tree (DT), k-nearest neighbors (KNN) and Random Forest (RF) classifiers. An ensemble voting of the three best classifiers is also performed in the classification process. A 5-fold cross-validation (CV) technique where 80% of the dataset represents the training set and the remaining 20% represents the testing set along with 4-fold CV and 10 fold CV are used to evaluate the result separately. A pictorial representation of the identification process for best feature and classification method is shown in Fig. 5.

After the identification of the best features set and classification method, the performance of the linguistic features based classifiers are evaluated. Confusion matrix, a summary of prediction results in a classification problem, is used for computing precision, accuracy, recall and F-score. The classifiers are implemented using scikit-learn. The implementation is elaborated in the following Sect. 4.

⁶ <https://scikit-learn.org/stable/>.

⁷ <https://github.com/arosh/BM25Transformer>.

⁸ <http://www.numpy.org/>.

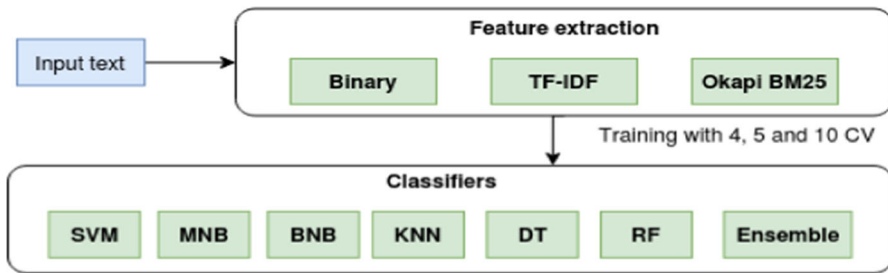


Fig. 5 Identification of best feature and classification

3.5 Classification using deep learning

In addition to the above classifiers, three deep learning based approaches, namely, Long Short Term Memory Networks (LSTM) (Hochreiter & Schmidhuber, 1997), Convolutional Neural Network (CNN) (Zhang & Wallace, 2015) and bidirectional Long Short-Term Memory (bLSTM) are also experimented. Unlike traditional machine learning techniques, deep learning algorithms are built on “artificial neural network”.

CNNs are made up of learnable weights and biases. CNN models extract higher-level features through convolutional layers and maximum pooling layers. In text classification, CNN can take care of words in short proximity but is unable to consider the context provided in a particular text sequence. LSTM models, on the other hand, specialize in long-term data retention by capturing long-term dependencies between word sequences and thus are better used for text classification. In addition, a bidirectional LSTM holds contextual data in both directions, which is helpful in the tasks of text classification.

4 Experimental result analysis and discussion

The experimental results and analysis are presented in this section. Sections 4.1 and 4.2 illustrate the experiments carried out using machine learning and deep learning techniques respectively.

4.1 Experiments using machine learning

Different machine learning classification methods were employed with the application of cross-validation for sentiment analysis. However, due to the sparseness of features, Gaussian Naive Bayes classifier cannot be evaluated. Experiments are carried out in two different steps.

- First, best feature set and classification method are identified by classifying the dataset into two classes.

- The best combination is then used to classify the dataset into two classes (i.e. negative and positive) and three classes (i.e. negative, positive and neutral) separately.

TfidfVectorizer and scikit-learn⁹ based classifier with their default settings are used for the experiment. Taking best feature set as a baseline, a comparison is carried out on the same feature set with higher weightage of negative morpheme-based lexicon. The annotated values of dataset *DI* are used as our gold standard.

4.1.1 Identification of best feature and classification

The classification experiments are carried out by taking equal number of negative and positive sentences to avoid biasness. Thus, 1138 sentences from the 1284 negative sentences and 1138 sentences from the 2587 positive sentences are randomly sampled. The total 2276 sentences are then shuffled and features were extracted using three methods separately: first according to the presence and absence of a term, i.e. binary values considered as a baseline reference, and second by using TF-IDF and finally using Okapi BM25. The features are then used separately for the classification process using the following classifiers: SVM, MNB, BNB, DT, KNN, RF, and ensemble voting (best three classifiers with the highest average result). The classification is done for 4-fold CV, 5-fold CV and 10-fold CV separately. Tables 4, 5 and 6 shows the result obtained for the classification into two classes, namely positive and negative with binary, TF-IDF and Okapi BM25 values respectively. The highest average overall result is obtained with TF-IDF values as the features in 10-fold CV. The ensemble voting classifier achieved better results than the other classifiers in almost all the cases. By taking the best combination from the above experiment: TF-IDF features with the ensemble voting in 10 fold CV, the performance of the linguistic features is evaluated.

4.1.2 Classification into two classes

With TF-IDF as the baseline feature, the classification into two classes is carried out and the result is compared with the TF-IDF values with higher weightage of negative markers (NW_1) as the additional linguistic features. With a balanced dataset of an equal number of positive and negative sentences, the classification is carried out using the ensemble voting with 10-fold CV. Table 7 shows the confusion matrix of the classification result. In Table 7, actual negative (*AN*) and actual positive (*AP*) represents the number of negative and positive sentences present in the experiment dataset respectively. While the predicted negative (*PN*) and predicted positive (*PP*) represents the number of negative and positive sentences predicted by the classifiers respectively. Table 8 shows the performance of the classification result computed using the values from the confusion matrix of Table 7 in terms of accuracy, precision, recall, and F-score. The feature with weighted

⁹ <https://scikit-learn.org/stable/>.

Table 4 Results of classifiers with baseline binary values

CV	SVM	MNB	BNB	KNN	DT	RF	Ensemble	APCV
4-fold	73.41	72.45	73.24	58.3	68.27	69.85	74.47	69.99
5-fold	73.5	74.47	74.29	57.73	69.85	70.07	74.56	70.63
10-fold	74.34	74.91	74.51	57.51	69.24	70.91	75.17	70.94
APC	73.75	73.94	74.01	57.84	69.12	70.27	74.73	

The bold text indicates that the results are higher ascompared to other models

Acronyms: APC = Average per Classifier, APCV = Average per CV

Table 5 Results of classifiers with TF-IDF values

CV	SVM	MNB	BNB	KNN	DT	RF	Ensemble	APCV
4 fold	74.2	73.37	74.6	65.55	66.52	67.53	74.95	70.96
5 fold	73.33	73.37	74.82	65.59	67.17	68.27	75.04	71.08
10 fold	74.64	73.72	75.08	65.11	67.92	68.67	75.57	71.53
APC	74.05	73.48	74.83	65.41	67.2	68.15	75.18	

Table 6 Results of classifiers with Okapi BM25 values

CV	SVM	MNB	BNB	KNN	DT	RF	Ensemble	APCV
4-fold	72.27	69.15	73.72	57.38	66.82	67.31	73.72	68.62
5-fold	72.89	71.44	74.34	55.71	67.97	68.23	73.94	69.21
10-fold	73.19	71.26	74.64	57.24	68.14	70.47	74.69	70.11
APC	72.78	70.61	74.23	56.77	67.64	68.67	74.11	

Table 7 Confusion matrix of the classification into two classes

	(a) Baseline			(b) Weighted			
	PN	PP	Total	PN	PP	Total	
AN	846	292	1138	AN	875	263	1138
AP	264	874	1138	AP	218	920	1138

Acronyms: AN = Actual Negative, AP = Actual Positive, PN = Predicted Negative, PP = Predicted Positive

negative markers shows improvement in terms of accuracy, recall, precision, and F-score as compared to the baseline.

Table 8 Results of baseline feature and weighted negative morpheme based lexicon

Feature	Accuracy	Precision	Recall	F-score
Baseline	75.57	76.8	74.95	75.86
Weighted	78.86	80.84	77.76	79.27

The bold text indicates that the results are higher ascompared to other models
Numerical values in percentage

Table 9 Confusion matrix of the classification into three classes

(a)Baseline					(b)Weighted				
	PNg	PNu	PP	Total		PNg	PNu	PP	Total
ANg	814	169	155	1138	ANg	821	173	144	1138
ANu	243	572	323	1138	ANu	165	609	364	1138
AP	245	304	589	1138	AP	159	327	652	1138

Acronyms: AP = Actual Positive, ANu = Actual Neutral, ANg = Actual Negative, PP = Predicted Positive, PNu = Predicted Neutral, PNg Predicted Negative

Table 10 Evaluation of baseline feature and weighted negative morpheme based lexicon in three class classification

Feature	Precision	Recall	F-score	Feature	Precision	Recall	F-score
(a) For negative class				(b)For neutral class			
Baseline	71.52	62.51	66.72	Baseline	50.26	54.73	52.4
Weighted	72.14	71.7	71.92	Weighted	53.51	54.91	54.2
(c) For positive class				(d)Average			
Baseline	51.75	55.2	53.42	Baseline	57.85	57.48	57.51
Weighted	57.29	56.2	56.74	Weighted	60.98	60.94	60.95

The bold text indicates that the results are higher ascompared to other models
Numerical values in percentage

4.1.3 Classification into three classes

A similar experiment is also carried out for the classification into three classes: negative, positive, and neutral. By taking a balanced dataset of positive, negative, and neutral polarity sentences and classifying it with the same baseline and weighted features. Table 9 shows the confusion matrix of the classification result. In Table 9, actual positive (AP), actual neutral (ANu) and actual negative (ANg) represents the number of positive, neutral, and negative sentences present in the experiment dataset respectively. While predicted positive (PP), predicted neutral (PNu) and predicted negative (PNg) represents the number of positive, neutral, and negative sentences predicted by the classifiers respectively. Table 10 shows the precision, recall, and F-score in terms of percentage for the classification result. In

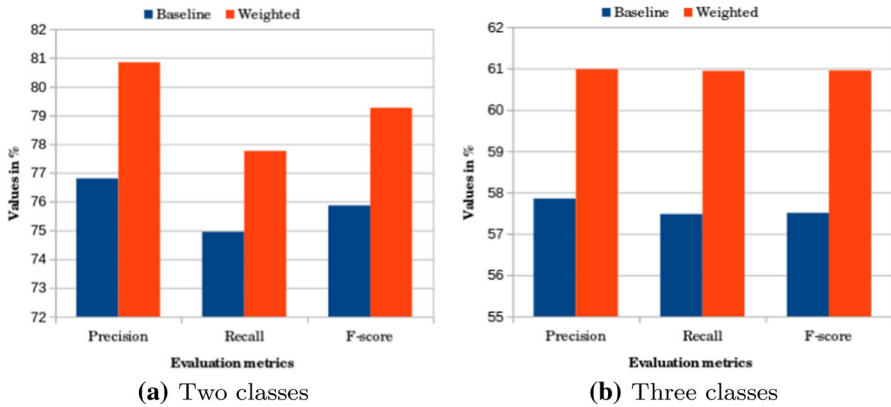


Fig. 6 Classification result comparison of the baseline and weighted negative morpheme based lexicon

Table 11 Result of the text predicted using ensemble voting classifier

Positive	Negative	Neutral	Total
43630	18,207	32,922	94,759

Table 12 Aggregate result of dataset classification

Dataset	Positive	Negative	Neutral
D1	2587	1284	1138
D2	43,630	39,094 (E+P)	32,922
Total	46,217	40,378	34,060

Table 13 Dataset partitioning statistics

	Sentences	Tokens
train-val	4509	1284
test	500	39,094
Total	5009	40,378

the case of three class classification, an improvement is observed when the negative marker words are taken as additional features over the baseline in terms of precision, recall, and F-score.

Figure 6 shows the summarized comparison of the classification results in the presence and absence of linguistic features for two classes and three classes. The chart in Fig. 6 is plotted by taking the performance parameter Precision, Recall, and

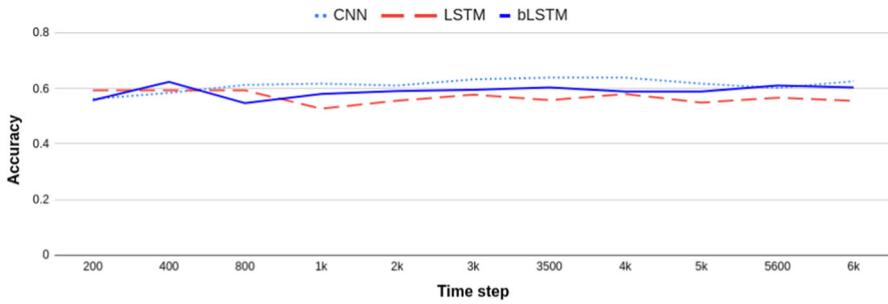


Fig. 7 Validation results in terms of accuracy at different time steps

Table 14 Evaluation on test dataset

	CNN	LSTM	bLSTM
Accuracy	60	58.48	62

F-score, which is grouped according to the type of features, in the x-axis and their values in terms of percentage in the y-axis.

4.1.4 Dataset D2

Additional experiments on dataset *D2* are carried out by extracting the list of sentences based on the presence of any terms in NM_2 . A total of 20887 negative sentences (*E*) are extracted from 115646 sentences. This rule-based approach of using negative markers fail to classify several sentences that express the negative sentiment but does not contain negative markers. Further, the rest of the dataset is classified through an ensemble voting classifier. The ensemble voting classifier used in the classification of dataset *D1* is applied on the remaining 94759 sentences of dataset *D2* to classify into positive, negative, or neutral. Table 11 shows the predicted (P) results of dataset *D2*. Table 12 summarizes the categorization result of dataset *D1* and dataset *D2*.

4.2 Experiments using deep learning

Additional experiment on dataset *D1* is carried out for the sentiment analysis using LSTM, CNN and bLSTM. This dataset is split into training and testing datasets as shown in Table 13.

With an embedding size of 256, filter size (3, 4, 5), learning rate of 0.001, and L2 regularization ($\lambda=0.001$), the CNN based model is trained for 6500 time steps. As for LSTM and bLSTM, the systems are trained separately for 6500 time steps each with an embedding size of 128, learning rate of 0.001, and L2 regularization ($\lambda=0.001$). Checkpoints are saved for each 100-time step in all

the systems. The checkpoint with the highest validation accuracy is then used to evaluate the test dataset. Figure 7 shows the validation result of the three models in terms of accuracy. Table 14 shows the evaluation result of the trained systems on test dataset.

5 Conclusions

We collected and prepared a goal standard dataset for Manipuri sentiment analysis from a local daily newspaper. Transliteration systems are implemented to transliterate Bengali script text to Roman script text and Meetei Mayek script text to Roman script text.

The 10-fold cross-validation shows better performance than the 4-fold and 5-fold cross-validation. Different classifiers with different types of vectors: TF-IDF and Okapi BM25 are experimented on the same dataset. Our experimental result indicates that TF-IDF outperforms the binary representation and Okapi BM25 ranking function. TF-IDF feature injection helped to improve the overall classification performance. Further, it is observed that ensemble voting of the best three classifiers achieved higher results compared to other classifiers in multi-class classification.

We also took advantage of morphological features of the Manipuri where the negative sentiment of a sentence or phrase could be identified by using negative markers. Improvement in the result was observed when the negative morpheme-based lexicons were given more weightage in the classification of two and three classes. However, the overall accuracy drops in the case of the classification into three classes. The reason being a misclassification due to inadequate features for all three classes. We have also acquired a dataset of negative morpheme-based lexicons for the Manipuri language which could be helpful for certain natural language processing tasks in the future.

In addition, we also carried out an evaluation of Manipuri sentiment analysis based on deep learning techniques namely, LSTM, CNN, and bLSTM. Our finding shows bLSTM to outperform CNN and LSTM in terms of accuracy on the test dataset.

Limited availability of good language-specific toolkits for Manipuri language acts as a major constraint and restricts the current work from incorporating additional linguistic features. Our transliterated gold standard dataset could be of use in extending the work on the dataset collected from social media with proper normalization.

Acknowledgements We would like to express our appreciation to L. Arunkumar's team and Preeti Q. Sinam for their assistance without whom the compilation of this corpus would not have been possible. We also thank the anonymous reviewers for their careful reading and their many insightful comments, which helped us to improve our manuscript.

References

- Albayati, A. Q., Al-Araji, A. S., & Ameen, S. H. A Method of Deep Learning Tackles Sentiment Analysis Problem in Arabic Texts.
- Cambria, E., & Hussain, A. (2015). SenticNet. In *Sentic Computing* (pp. 23–71). Springer, Cham.
- Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017). Affective computing and sentiment analysis. In *A practical guide to sentiment analysis* (pp. 1–10). Springer, Cham.
- Das, A., & Bandyopadhyay, S. (2010, August). SentiWordNet for Indian languages. In *Proceedings of the Eighth Workshop on Asian Language Resources* (pp. 56–63).
- Dashtipour, K., Gogate, M., Li, J., Jiang, F., Kong, B., & Hussain, A. (2020). A hybrid Persian sentiment analysis framework: Integrating dependency grammar based rules and deep neural networks. *Neurocomputing*, 380, 1–10.
- Denecke, K. (2008, April). Using sentiwordnet for multilingual sentiment analysis. In *2008 IEEE 24th International Conference on Data Engineering Workshop* (pp. 507–512). IEEE.
- El-Haj, M., Kruschwitz, U., & Fox, C. (2015). Creating language resources for under-resourced languages: methodologies, and experiments with Arabic. *Language Resources and Evaluation*, 49(3), 549–580.
- Gangula, R. R. R., & Mamidi, R. (2018, May). Resource creation towards automated sentiment analysis in telugu (a low resource language) and integrating multiple domain sources to enhance sentiment prediction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Goldberg, A. B., & Zhu, X. (2006, June). Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the first workshop on graph based methods for natural language processing* (pp. 45–52). Association for Computational Linguistics.
- Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17, 26–32.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Hu, X., Tang, J., Gao, H., & Liu, H. (2013, May). Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 607–618). ACM.
- Jang, H., & Shin, H. (2010, August). Language-specific sentiment analysis in morphologically rich languages. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 498–506). Association for Computational Linguistics.
- Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, 5, 2870–2879.
- Johansson, F., Brynielsson, J., & Quijano, M. N. (2012, August). Estimating citizen alertness in crises using social media monitoring and analysis. In *2012 European Intelligence and Security Informatics Conference* (pp. 189–196). IEEE.
- Khan, A., & Baharudin, B. (2011, September). Sentiment classification using sentence-level semantic orientation of opinion terms from blogs. In *2011 National Postgraduate Conference* (pp. 1–7). IEEE.
- Kim, S. M., & Hovy, E. (2004, August). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics* (p. 1367). Association for Computational Linguistics.
- Le, T. A., Moeljadi, D., Miura, Y., & Ohkuma, T. (2016, December). Sentiment analysis for low resource languages: A study on informal Indonesian tweets. In *Proceedings of the 12th Workshop on Asian Language Resources (ALR12)* (pp. 123–131).
- Lo, S. L., Cambria, E., Chiong, R., & Cornforth, D. (2017). Multilingual sentiment analysis: From formal to informal and scarce resource languages. *Artificial Intelligence Review*, 48(4), 499–527.
- Mishne, G. (2005, August). Experiments with mood classification in blog posts. In *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access* (Vol. 19, pp. 321–327).
- Na, J. C., Sui, H., Khoo, C. S., Chan, S., & Zhou, Y. (2004). Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. *International ISKO Conference*.

- Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70–77). ACM.
- Niu, Y., Zhu, X., Li, J., & Hirst, G. (2005). Analysis of polarity information in medical text. In *AMIA annual symposium proceedings* (Vol. 2005, p. 570). American Medical Informatics Association.
- Nongmeikapam, K., Khangembam, D., Hemkumar, W., Khuraijam, S., & Bandyopadhyay, S. (2014). Verb based manipuri sentiment analysis. *International Journal on Natural Language Computing (IJNLC)*, 3, 12–13.
- Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc* (Vol. 10, No. 2010, pp. 1320–1326).
- Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, Vol. 10 (pp. 79–86). Association for Computational Linguistics.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), 2825–2830.
- Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. *Nist Special Publication Sp*, 109, 109.
- Singh T.D., Singh T.J., Shadang M., & Thokchom S. (2021) Review Comments of Manipuri Online Video: Good, Bad or Ugly. In: Maji A.K., Saha G., Das S., Basu S., Tavares J.M.R.S. (eds) *Proceedings of the International Conference on Computing and Communication Systems*. Lecture Notes in Networks and Systems, vol 170. Springer, Singapore.
- Singh, T. D. (2012, December). Bidirectional bengali script and meetei mayek transliteration of web based manipuri news corpus. In *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing* (pp. 181–190)
- Singh, T. D., & Bandyopadhyay, S. (2006). Word class and sentence type identification in manipuri morphological analyzer,” In *Proceedings of MSPIL*, Mumbai, India, 11–17.
- Singh, T. D., & Bandyopadhyay, S. (2010, August). Web Based Manipuri Corpus for Multiword NER and Reduplicated MWEs Identification using SVM. In *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing* (pp. 35–42).
- Sixto, J., Almeida, A., & López-de-Ipiña, D. (2016, June). Improving the sentiment analysis process of Spanish Tweets with BM25. In *International Conference on Applications of Natural Language to Information Systems* (pp. 285–291). Springer, Cham.
- Sixto, J., Almeida, A., & López-de-Ipiña, D. (2016, September). An approach to subjectivity detection on Twitter using the structured information. In *International Conference on Computational Collective Intelligence* (pp. 121–130). Springer, Cham.
- Sixto, J., Almeida, A., & López-de-Ipiña, D. (2018). Analysis of the Structured Information for Subjectivity Detection in Twitter. In *Transactions on Computational Collective Intelligence XXIX* (pp. 163–181). Springer, Cham.
- Vilares, D., Peng, H., Satapathy, R., & Cambria, E. (2018, November). BabelSenticNet: a commonsense reasoning framework for multilingual sentiment analysis. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1292–1298). IEEE.
- Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.
- Yu, H., & Hatzivassiloglou, V. (2003, July). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 129–136). Association for Computational Linguistics.
- Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. [arXiv:1510.03820](https://arxiv.org/abs/1510.03820).
- Zhang, W., Xu, H., & Wan, W. (2012). Weakness finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis. *Expert Systems with Applications*, 39(11), 10283–10291.