SentNoB: A Dataset for Analysing Sentiment on Noisy Bangla Texts

Khondoker Ittehadul Islam[⋆], **Md Saiful Islam**[⋆‡], **Sudipta Kar**[†] **and Mohammad Ruhul Amin**[◇]
[⋆]Shahjalal University of Science and Technology, Bangladesh
[‡]University of Alberta, Canada, [†]Amazon Alexa AI, USA, [◇]Fordham University, USA
khondoker07@student.sust.edu, mdsaiful@ualberta.ca
sudipkar@amazon.com, mamin17@fordham.edu

Abstract

In this paper, we propose an annotated sentiment analysis dataset made of informally written Bangla texts. This dataset comprises public comments on news and videos collected from social media covering 13 different domains, including *politics*, *education*, and *agriculture*. These comments are labeled with one of the polarity labels, namely *positive*, *negative*, and *neutral*. One significant characteristic of the dataset is that each of the comments is noisy in terms of the mix of dialects and grammatical incorrectness. Our experiments to develop a benchmark classification system show that hand-crafted lexical features provide superior performance than neural network and pre-trained language models. We have made the dataset and accompanying models presented in this paper publicly available at https://git.io/JuuNB.

1 Introduction

Sentiment analysis is one of the classic problems in computational linguistics, and it has shown a massive impact on different real-life applications. The capability to quantify sentiment polarity of English texts has enabled the creation of solutions for a diverse set of problems like understanding the possible movement of stock markets, public sentiment towards any event or product, and understanding client satisfaction for customer support. A major reason behind such a success is the amount of collaborative efforts invested in the research and development of the creation of public resources like Sentiment140 (Go et al., 2009; Mohammad et al., 2013), SentiWordNet (Baccianella et al., 2010), IMDB review corpus (Maas et al., 2011), Stanford Sentiment Treebank (Socher et al., 2013), TS-Lex (Tang et al., 2014), and SemEval Twitter sentiment analysis corpus (Rosenthal et al., 2017).

Positive	[**B**] অ অ অ অ সাধারন । আমি কোন দিনই পারবো না । হিংসা হচ্ছে [**E**] *Great. I will never be able to do it. Feeling jealous.*
Neutral	[**B**] পিছনে দুজন মুর্তি দারা করায় লাগছে [**E**] *Two people placed idols behind them.*
Negative	[**B**] ভাই আপনার ক্যামেরা মেনকে দিলেন্না একাই সব সাবার করলেন, হা হা হা [**E**] *Bro, you didn't share with your cameraman and ate the whole thing, Ha Ha Ha.*

Table 1: Samples from our dataset with each demonstrating certain challenges. **B** represents the original instance in Bangla and **E** is its English translation.

Bangla is the sixth most spoken language worldwide and the second Indo-Aryan language after Hindi (Eberhard et al., 2021)[1] with 268M speakers. Bangla is the native language of Bangladesh and some regions of India, such as West Bengal. While technology is dramatically improving the lives of people from these densely populated and economically burgeoning regions, it is a timely need of building technologies that can understand the language, enhancing the overall impact on social welfare and businesses.

Existing datasets for sentiment analysis for a low-resource language like Bangla suffer from three major limitations: 1) none to slight inter annotator agreement score questioning the annotation reliability (e.g., 0.11 in Ashik et al., 2019 and 0.18 in Islam et al., 2020), 2) lack of cross-domain generalization capability due to large domain dependency (Wahid et al., 2019; Rahman et al., 2019; Sazzed, 2020), and 3) lack of public availability for further research (Karim et al., 2020; Nabi et al., 2016; Hassan et al., 2016; Sharmin and Chakma, 2020; Choudhary et al., 2018; Das and Bandyopadhyay, 2009).

In this paper, we aim at creating a domain-representative sentiment polarity classification

[1] https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

dataset by collecting public opinions on various topics. During the data collection and annotation process, we invest efforts to improve the quality of the dataset using data curation techniques. On one hand, it includes the steps for duplicate removal, while on the other hand we increase the vocabulary size by incorporating instances that will help to increase the unique word percentage. Our contributions can be summarized as follows:

- We propose SentNoB, a dataset for analysing **Sent**iment in **No**isy **B**angla texts. This dataset is a collection of ≈15K social media comments on news and videos from 13 different domains. Instances from the dataset demonstrate heavy usage of different local dialects, spelling, and grammatical errors. We show some examples in Table 1.
- We experiment on different techniques such as linguistic features, recurrent neural networks, and pre-trained language model; and show that old-school lexical features like word n-grams demonstrate superior performance in classification. We shed light on different aspects of the problem throughout our analysis.
- We make our dataset and model publicly available to foster research in this direction.

2 Development of SentNoB

Data Collection We defined the following objectives before creating the dataset as we believe these objectives will enhance the generalization capability of SentNoB: 1) Samples should represent many different domains to encourage domain-independent solutions. 2) Samples should contribute to making the dataset less repetitive. We start by collecting public comments on articles on the most popular 13 topics from Prothom Alo[2], the most circulated newspaper in Bangladesh[3]. Then we collect comments from a set of Youtube videos on similar topics.

Out of ≈ 31K collected comments, we keep the comments that are written in only Bangla alphabets. To reduce repetitiveness and noise, we remove duplicates and exclude instances shorter than three or longer than 50 words tokens. Additionally, we aim at increasing the vocabulary size by incorporating as many different words as possible. Therefore, we prioritize the instances for annotation that will increase the percentage of the unique word in the dataset. Diverse vocabulary poses a challenge in modeling but eventually helps to create more robust classification systems that can generalize well.

Class	Instances	#Sent/instance	#Words/instance
Negative	5,709 (36.3%)	1.64	16.33
Positive	6,410 (40.8%)	1.73	15.88
Neutral	3,609 (22.9%)	1.45	12.94
Total	15,728	1.63	15.37

Table 2: Brief statistics of SentNoB per class label.

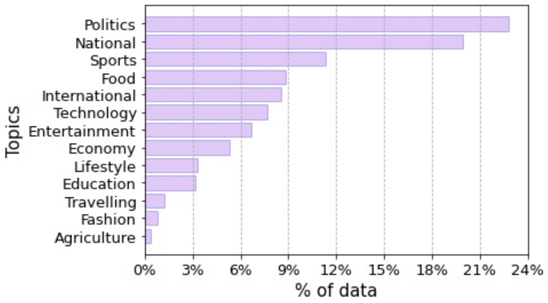

Figure 1: Topic distribution of the dataset.

Annotation We use three different annotators to label each instance with one of the five polarity labels *Strong Negative, Moderate Negative, Neutral, Moderate Positive*, and *Strong Positive*. For this task, we employed ten undergraduate students and provided them with detailed annotation guidelines. We use majority voting to assign the final class label, where we keep the neutral class unchanged but combine the two intensities of the polar classes and assign either *Positive* or *Negative* label. An inter-annotator agreement (Fleiss, 1971) score of 0.53 indicates a moderate agreement across the dataset. To our knowledge, this is the highest such score among the Bangla datasets that made the agreement score public.

Statistics and Analysis In total, we have $15,728$ instances in the final dataset (Table 2). The average length of the instances is 1.63 ± 1.03 sentences and average sentence length is 15.37 ± 9.93 words. 40.8% of the data are labeled as *Positive*, 36.3% *Negative*, and 22.9% *Neutral*. Figure 1 shows the topic distribution of the dataset. While, 42.73% instances are from *national* and *political* news, we have less data from *fashion* and *agriculture*.

We observe that agreement decreases with in-

[2] https://www.prothomalo.com

[3] https://www.top10bd.com/top-10-newspaper-in-bangladesh

stance length. For instance, all three annotators agreed for 36% texts with 11-20 tokens, 15.07% texts with 21-30 tokens, and 7.08% texts for 31-40 tokens. This is intuitive as longer texts can pose sentiment contradiction among different segments and often challenge annotators' own biases and perspectives. For example, we observe low agreements on data from *politics* and *national* domain as these domains demonstrate heavy partisanship.

3 Methodology

In this section, we describe the methods we investigate to develop a benchmark model for classifying sentiment polarity on SentNoB. We start by training linear SVM (Cortes and Vapnik, 1995) models with traditional hand-engineered linguistic features. Then, we experiment with recurrent neural network models and pre-trained transformer based language models due to their recent success on a wide variety of NLP tasks.

3.1 Linguistic Features

Lexical We extract word (1-3) and character (2-5) n-grams from the instances as these lexical representations have shown strong performance in different classification tasks. Then we vectorize each instance with the TF-IDF weighted scores for each n-gram.

Semantic To utilize semantic information from the texts, we experiment with FastText (Grave et al., 2018) pre-trained Bangla word embeddings, where we represent a text with the mean of the vectors for each word. FastText has 81.75% coverage on our dataset as FastText's training data are formal Bangla texts from Wikipedia, whereas we created our dataset with informal Bangla texts written by general people on the internet. We considered FastText embedding for linguistic feature-based experiments. We represent the out of vocabulary words with zero vector.

3.2 Recurrent Neural Networks

We use a bidirectional long short-term memory (BiLSTM; Hochreiter and Schmidhuber, 1997) network that encodes a text from the forward and backward directions and creates a 2D vector for each direction. Then, we concatenate the vectors and apply attention mechanism (Bahdanau et al., 2015) that learns to put more weight on the words crucial for correct classification. We compute the attention weighted sum of the vectors and predict the sentiment polarity through an output layer. Instead of using any pre-trained embeddings (e.g., FastText) to initialize the embedding layer, we use random initialization because of better performance in some initial experiments.

3.3 Pre-trained Language Model

In recent years, large pre-trained language models like BERT (Devlin et al., 2018) have shown impressive performance in a wide range of linguistic tasks of many languages. Therefore, we assess the performance of such a model by fine-tuning it on our dataset. We choose the multi-lingual BERT (mBERT) as its training data included Bangla texts, and only fine-tune the output layer with our training data due to computing resource limitation.

4 Experimental Setup

We implement our experimental framework using Pytorch (Paszke et al., 2019), Transformers (Wolf et al., 2020), and Scikit-learn (Pedregosa et al., 2011). We evaluate our methods using micro averaged F1. As the baseline systems, we compare our results with the majority, random, and weighted random baselines. To reduce noise, we replace the numerical tokens with a CC token and normalize English and Bangla sentence stoppers. Due to the class imbalance, we perform per-topic stratified split to create training (80%), development (10%), and test (10%) sets.

While we evaluate all the individual features using the same hyper-parameter setting, we tune the SVM regularizer C[4] of the model on the validation set performance for the best performing feature combination. For training the BiLSTM model with mini batches, we left pad the instances and perform hyper-parameter tuning on learning rate, batch size, dropout rate, number of LSTM cells and layers. For fine-tuning mBERT, we only tune the learning rate and batch size.

5 Results and Analysis

We report our experimental results on the test set in Table 3. The majority baseline achieves a 41.24 F1 score by assigning the dominant label ($+ve$) to every instance, which is better than the random baselines (34.53 and 32.60). Among the word n-grams, we observe better performance with unigram 63.19 compared to bigram (59.68) and trigram (55.56).

[4] We tested on these values: $1e^{-3}$, $1e^{-2}$, 0.1, 1 (best), 10.

Method	Precision	Recall	F1
Majority	41.24	41.24	41.24
Random	33.67	35.44	34.53
Weighted Random	31.89	33.35	32.60
Bi-LSTM + Attn. (FastText)	52.24	63.09	57.15
Bi-LSTM + Attn. (Random)	56.16	64.97	60.25
mBERT	49.58	56.43	52.79
Unigram (U)	56.89	71.06	63.19
Bigram (B)	54.32	66.20	59.68
Trigram (T)	51.57	60.21	55.56
U + B	**57.71**	72.95	64.44
U + B + T	57.03	71.88	63.60
Char 2-gram (C2)	53.29	66.39	59.12
Char 3-gram (C3)	56.06	70.87	62.60
Char 4-gram (C4)	56.62	71.44	63.17
Char 5-gram (C5)	56.94	71.94	63.57
C2 + C3	56.00	70.93	62.59
C3 + C4	56.49	71.31	63.04
C4 + C5	57.30	72.76	64.11
C2 + C3 + C4	56.45	71.44	63.07
C3 + C4 + C5	57.60	73.39	64.54
C2 + C3 + C4 + C5	57.06	72.89	64.01
U + B + C3 + C4 + C5	56.96	72.51	63.80
U + B + C2 + C3 + C4 + C5	57.05	72.70	63.93
U + B + T + C2 + C3 + C4 + C5	**57.71**	**73.39**	**64.61**
Embeddings (E)	50.68	63.75	56.46
U + B + C2 + C3 + C4 + C5 + E	57.48	73.14	64.37
U + B + T + C2 + C3 + C4 + C5 + E	57.36	72.45	64.03

Table 3: Precision, Recall, and F1 for different methods.

Combining bigram with unigram lifts the unigram F1 by 1.25 (i.e., 64.44), but adding trigram to that combination reduces the rate of improvement, and we achieve 63.60 F1. We observe similar classification performance with the character n-grams.

While character 3, 4, and 5 grams' performances are around 3-4% higher than character bigram, the difference among their F1 scores is low. Surprisingly, different combinations of the character n-grams do not show significantly higher gains. Combining all character n-grams yields a small gain of 0.44 over the most robust character 5-gram feature. However, we do not observe any significant shift in the precision and recall scores for character n-gram combinations. This implies that the task highly depends on word units and does not rely much on the subword level information. Integrating the all word n-grams with all character n-grams achieves the best F1 of 64.61, and improves on both precision and recall. The embedding feature demonstrates poor performance (F1=56.46), and combining this with the lexical features does not show any improvement.

According to our results, linguistic feature combinations perform better than the neural models on our dataset. Although the Bi-LSTM model's precision is closer to the precision of the lexical feature combination approach, the recall is ≈ 8% lower (64.97 vs 73.39). We observe that mBERT's performance (F1=52.79) is significantly lower than the Bi-LSTM model.

Train	Test	Precision	Recall	F1
Informal	Informal	37.29	44.00	40.37
Formal	Formal	40.00	44.00	41.90
Formal	**Informal**	40.32	**50.00**	**44.64**
Informal	Formal	**41.07**	46.00	43.40

Table 4: Results of few-shot experiments with different train-test combinations of formal and informal texts. The best

There can be two possible reasons behind such a performance: a) mBERT's training data is compiled of formal Bangla text from Wikipedia, whereas our dataset contains informal and noisy Bangla texts, and b) fine-tuning only the output layer makes mBERT under-trained for the task. To verify the first hypothesis, we randomly sample 100 instances from the training and validation sets, and manually translate them to formal Bangla. Then, we perform some few-shot experiments on mBERT with different train-test combinations of the formal and informal versions. Although the dataset for this experiment is very small, the results in Table 4 indicates that the first hypothesis is not true. If the hypothesis was true, we would have observed the best performance when both training and test sets are made of formal texts. But, the results are quite the opposite. Best F1 is achieved when the training material is formal text, but test set is informal text. This suggests that fine-tuning only the output layer of mBERT probably leaves the model under-trained for this task. However, poor performance of FastText embeddings (pre-trained on Wikipedia) than random embeddings in BiLSTM model adds some support towards the first hypothesis. In the future, we plan to further investigate in this direction.

Performance by Topic Analysing the results per topic and per class from Table 5, we find that the F1 difference for *+ve* and *-ve* class is small (78.99 vs 76.29), but 42.25 F1 indicates that the *Neutral* samples are the hardest to identify. F1 for the Negative class is comparatively higher for topics like *Politics* and *Economy* as ideological conflicts are mostly responsible for negativity in these topics. Additionally, we find that people tend to speak more about their negative experiences about *Food, Travel,* and *Tech* products, and our approach shows higher recall in these topics. Interestingly, *+ve* instances are harder to identify for *Tech*. Although

Category	Support	Neutral				Positive				Negative				F1	F.A.
		S	P	R	F1	S	P	R	F1	S	P	R	F1		
Politics	360	88	69.49	46.59	55.78	145	75.32	80.00	77.59	127	74.15	85.83	79.56	64.56	59.72
National	314	73	61.90	35.62	45.22	135	76.32	85.93	80.84	106	73.33	83.02	77.88	64.97	62.42
Sports	178	43	55.17	37.21	44.44	70	73.75	84.29	78.67	65	75.36	80.00	77.61	63.66	52.81
Food	140	31	**73.91**	54.84	62.96	55	72.73	72.73	72.73	54	74.19	85.19	79.31	63.58	60.71
International	136	31	65.38	54.84	59.65	60	76.56	81.67	79.03	45	76.09	77.78	76.92	66.67	45.59
Tech	122	25	43.75	56.00	49.12	55	**86.05**	67.27	75.51	42	74.47	83.33	78.65	62.55	66.39
Entertainment	106	22	55.00	55.00	52.38	41	76.60	87.80	81.82	43	87.18	79.07	82.93	67.22	63.21
Economy	85	17	66.67	35.29	46.15	37	76.32	78.38	77.33	31	73.68	**90.32**	81.16	63.96	69.41
Lifestyle	53	13	50.00	23.08	31.58	14	61.11	78.57	68.75	26	68.97	76.92	72.73	54.84	50.94
Education	51	10	42.86	30.00	35.29	26	84.00	80.77	82.35	15	63.16	80.00	70.59	64.29	41.18
Travel	20	4	60.00	**75.00**	**66.67**	7	75.00	85.71	80.00	9	**100.0**	77.78	**87.50**	**69.57**	60.00
Fashion	14	3	00.00	00.00	00.00	5	71.43	**100.0**	83.33	6	71.43	83.33	76.92	60.61	78.57
Agriculture	7	1	00.00	00.00	00.00	4	80.00	**100.0**	**88.89**	2	50.00	50.00	50.00	66.67	71.43
Avg.			49.55	38.73	42.25		75.78	83.32	78.99		74.00	78.66	76.29	64.09	60.18

Table 5: Support (S), Precision (P), Recall (R), and F1 score for each topic per class label. The F.A. column indicates the percentage of training samples where all three annotators agreed on the class label.

Positive: ধন্যবাদ (thanks), অসাধারণ (great), মেধাবী (talented), খুব ভাল (very good), বেস্ট ! (best !), রিপোর্টটা অসাধারণ চিল (the report was great), আলহাদুলিল্লাহ গ্রেট নিউজ (thanks god great news), ❤ ❤ ❤
Negative: পুলিশ (police), হনুমান (monkey), বালের (slang), খুন (murder), ধিক্কার (indignation), জবাই (slaughter), কুত্তার বাচ্চা (slang), বিচার হবে না (there will be no justice), মেরে ফেলা উচিৎ (should be killed), গরিবেরা সবজায়গায় নিপীড়িত (the poor are oppressed everywhere)
Neutral: ফোন (phone), প্রাইভেট (private), আলোচনা (discussion), রাষ্ট্রপতি (president), প্রশ্ন (question), না ভাইয়া (no brother), ঠিক বলেছেন (you are right), বুঝলাম না কিছুই (didn't understand anything), টাকা লাগে না (it doesn't cost money)

Table 6: Examples of some of the strongest word n-grams from each class with their English translations.

we have a very small amount of data for *Education, Fashion* and *Agriculture*, *+ve* class's performance is significantly higher for these topics.

Dominant Features Table 6 shows some of the strongest n-gram features from each class. We observe that n-grams expressing strong positive emotions and compliments act as the indicator of the positive class, and they are mostly adjectives. On the other hand, negative samples are often associated with *police*, *crime*, *lack of trust in the judicial system*, and *slang*. Strongest n-grams for the neutral class are mostly nouns or information. We notice that many of the strongest n-grams are misspelled. Therefore, we believe pre-processing techniques like spell-correction and word segmentation can help normalize such noises and help to get better performance.

6 Conclusion

In this paper, we present SentNoB, a dataset for analysing sentiment in noisy Bangla texts collected from the comments section of Bangla news and videos from 13 different domains. SentNoB contains $\approx$ 15K instances labeled with positive, negative, or neutral class label. We found that lexical feature combinations demonstrate stronger classification performance compared to neural models. As the future work, we will focus on different preprocessing techniques and more investigation with pre-trained language models.

References

Md Akhter-Uz-Zaman Ashik, Shahriar Shovon, and Summit Haque. 2019. Data set for sentiment analysis on bengali news comments and its baseline evaluation. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5. IEEE.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nurendra Choudhary, Rajat Singh, Vijjini Anvesh Rao, and Manish Shrivastava. 2018. Twitter corpus of resource-scarce languages for sentiment analysis and multilingual emoji prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1570–1577.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Amitava Das and Sivaji Bandyopadhyay. 2009. Subjectivity detection in english and bengali: A crf-based approach. *Proceeding of ICON*.

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

David Ms Eberhard, David M, Gary F. Simons, and Charles D. Fennig. 2021. Languages of the world.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Asif Hassan, Mohammad Rashedul Amin, N Mohammed, and AKA Azad. 2016. Sentiment analysis on bangla and romanized bangla text (brbt) using deep recurrent models. *arXiv preprint arXiv:1610.00369*.

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

Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5.

Md Karim, Bharathi Raja Chakravarthi, Mihael Arcan, John P McCrae, Michael Cochez, et al. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. *arXiv preprint arXiv:2004.07807*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Muhammad Mahmudun Nabi, Md Tanzir Altaf, and Sabir Ismail. 2016. Detecting sentiment from bangla text using machine learning technique and feature analysis. *International Journal of Computer Applications*, 153(11).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Fuad Rahman, Habibur Khan, Zakir Hossain, Mahfuza Begum, Sadia Mahanaz, Ashraful Islam, and Aminul Islam. 2019. An annotated bangla sentiment analysis corpus. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5. IEEE.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.

Salim Sazzed. 2020. Cross-lingual sentiment classification in low-resource bengali language. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 50–60.

Sadia Sharmin and Danial Chakma. 2020. Attention-based convolutional neural network for bangla sentiment analysis. *AI & SOCIETY*, pages 1–16.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*, pages 172–182.

Md Ferdous Wahid, Md Jahid Hasan, and Md Shahin Alom. 2019. Cricket sentiment analysis from bangla text using recurrent neural network with long short term memory model. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–4. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush.

2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.