A Novel Approach towards Cross Lingual Sentiment Analysis using Transliteration and Character Embedding

Rajarshi Roychoudhury[1], Subhrajit Dey[1], Md Shad Akhtar[2]
Amitava Das[3], Sudip Kumar Naskar[1]
[1]Jadavpur University, India
[2]IIIT-Delhi, India
[3]AI Institute, University of South Carolina, Columbia, SC, USA
{rroychoudhury2,subhrajitdey.agt,sudip.naskar}@gmail.com, shad.akhtar@iiitd.ac.in, amitava@mailbox.sc.edu

Abstract

Sentiment analysis with deep learning in resource-constrained languages is a challenging task. In this paper, we introduce a novel approach for sentiment analysis in resource-constrained scenarios using character embedding and cross-lingual sentiment analysis with transliteration. We use this method to introduce the novel task of inducing sentiment polarity of words and sentences and aspect term sentiment analysis in the no-resource scenario. We formulate this task by taking a metalingual approach whereby we transliterate data from closely related languages and transform it into a meta language. We also demonstrated the efficacy of using character-level embedding for sentence representation. We experimented with 4 Indian languages – Bengali, Hindi, Tamil, and Telugu, and obtained encouraging results. We also presented new state-of-the-art results on the Hindi sentiment analysis dataset leveraging our metalingual character embeddings.

1 Introduction

Sentiment analysis is a widely explored topic in the field of Natural Language Processing, which focuses on classifying text into 3 sentiment classes: positive, negative, and neutral (Liu, 2012). For any text classification task, supervised approaches require an extensive and domain specific corpus in the corresponding language. However, sentiment annotated data, which is the primary resource for sentiment analysis, is limited in many languages. Transfer learning can be used to learn sentence representations by pretraining on a large corpus and finetuning to a particular downstream task. But transfer learning often fails in adapting knowledge from one domain to another, and sufficient training samples across these domains may not be available for efficient finetuning. Another method to deal with such situations is to create cross lingual tools between a resource rich and resource poor language. This method either uses machine translation between the language or bilingual dictionaries to overcome the language gap. However, it is an extremely challenging task to create an accurate machine translation system between two languages or create a bilingual dictionary pair in a resource constrained scenario (Balamurali et al., 2012). Moreover, there are some studies (Lohar et al., 2017, 2018; Pal et al., 2014; Kumari et al., 2021) that show that there is a significant loss of pragmatics in the translations produced by the state-of-the-art Machine Translation (MT) systems, which could adversely affect the performance of these downstream NLP applications that use unedited raw MT output. Very little work has been done to use cross-lingual sentiment analysis without translation involved in any step, and even if they do, they require the presence of large sentiment annotated data in at least one language.

Another major problem is the unavailability of significantly large corpus to train the language-specific word embedding models in resource scarce languages. As a result, training word embeddings on those languages and reusing the pretrained embeddings is not feasible. Literature suggests that using character embedding or subword embedding can be an efficient alternative, as the number of unique characters and alphabets for any language are well deterministic (Chrupała, 2013; dos Santos and Gatti, 2014; Mikolov et al., 2012; Kim, 2019). The main advantage of using character embedding is that when we encounter a word which is not in the domain of the available corpus (i.e., an out-of-vocabulary word, or OOV), we can still have a latent representation of the word for any NLP task. Previously, dos Santos and Gatti

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 260 - 268
December 15-18, 2022 ©2022 Association for Computational Linguistics

(2014) used character embedding for sentiment analysis of short texts, however, very few work have been done on word level classification.

There exist many such languages where the coverage of the corpus is too poor to effectively train the embedding weights for each word. Even transfer learning with pretrained large models like m-BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), Electra (Clark et al., 2020) might encounter many unknown ([UNK]) tokens since the vocabulary set of the tokenizers may not have many words in the test-set. Moreover, a dataset for the downstream task may not exist in the language for effective finetuning. Therefore, we need methods that can help create sentiment lexicons effectively and perform sentiment analysis without using any training dataset of the target language.

In this paper, we introduce a novel method to learn efficient latent representation by projecting data in multiple languages to a shared metadata representation. We propose to utilize the WX-notations as the transliteration approach for the projection. This enables us to leverage the collective training corpus to learn a deep learning system with higher confidence. We evaluate our approach on 7 sentiment analysis datasets (4 word-level, 2 sentence-level, and 1 aspect-level) across four Indian languages – Hindi, Telugu, Bengali, and Tamil. Moreover, we experiment in two setups – mono-lingual and cross-lingual. Finally, we compare our model against various baselines and observe the performance to be better for the majority of the cases. Convolutional Neural Network (CNN) provides the best result on aspect analysis and sentiment classification of sentences, while different models prove to be useful on different datasets for word-level sentiment analysis.

Contribution: The paper makes the following contributions.

- We introduce the novel task of predicting the sentiment polarity of words.
- Word-level and sentence level *zero-shot* sentiment analysis using a metalingual approach.
- Our proposed approach provides new state-of-the-art results on the $Review_{SH}$, $Review_{AH}$ and $Movie_H$ (Akhtar et al., 2016) Hindi sentiment analysis datasets.

2 Related Work

Wan (2009) introduced the method of cross-lingual sentiment analysis, which leveraged an available English corpus for Chinese sentiment classification by using the English corpus as training data without using any Chinese resources using a co-training approach. Zhou et al. (2016) proposed a joint learning algorithm that exploits both monolingual and bilingual constraints, where the monolingual constraints help to model words and documents in each language while the bilingual constraints help to build a consistent embedding space across languages. Abdalla and Hirst (2017) used cross sentiment analysis by computing a matrix to convert from the vector space of one language to that of another, based on the fact that that sentiment is highly "preserved" even if translation accuracy is poor. Here it is worth noting that most of these methods use a single resourceful language and use it for a resource-scare language, what sets us apart is that we have used multiple source languages and combined their resources to train hybrid embedding weights.

Rasooli and Collins (2017) combined a method for deriving cross-lingual clusters and a method for transfer of lexical information from the target language into source language treebanks with the density-driven approach to annotation projection for cross-lingual sentiment analysis on different source and destination languages instead of a single source language. Jain and Batra (2015) used Bilingually Constrained Recursive Auto-encoder (BRAE) (Zhang et al., 2014) to perform Cross-Lingual sentiment analysis. However, in most of the works cited above, we have to use translation to obtain cross-lingual relations between language pairs. As mentioned earlier, statistical machine translation is computation-intensive and demands an extensive corpus of bilingual text.

The main difficulty is cross alignments, due to word order/ syntactic differences in languages. Balamurali et al. (2012) presented an alternative approach to CLSA using WordNet senses as features for supervised sentiment classification. But for many resource-constrained languages, WordNet does not exist. Hence in the proposed methodology, we propose using transliteration as alignments are monotonic, ie, they do not cross each other. Previously an attempt was made to transliteration for sentiment analysis. In oder to automatically classify sentiments of Arabizi messages. Guellil et al. (2018) transliterated their corpus into Arabic and used classification models like Support Vector Machines, Naive Bayes classifier and Decision Trees for sentiment analysis. However for our experiments we will use a transliteration scheme to obtain a common metalanguage that can be used cross-lingual embedding training, and classify words/ sentences using simple classification models without using any data of the resource-scarce language.

3 Proposed Method

Training Deep neural network requires huge amount of data, which is not feasible for low resource languages. To avoid this, we follow the principles of cross-lingual learning, where model trained on one language is reused on another language. This ensures that no data from the target language is used during training, and hence can be adapted in a low resource scenario.

Our approach is as follows: similar sentiment-annotated datasets from other languages are leveraged to create a pool of data instead of a single source language data, which is subsequently transliterated into a common language (metalanguage) using a transliteration scheme. We use this combined metadata as our training data, which acts as a relatively bigger dataset that can be used for efficient training of deep learning-based text classification models. Additionally, it reduces the dependency on a single language dataset. The target language dataset is also transliterated using the same transliteration scheme and this transliterated data is used only as the testing data and plays no part in training.

We train our classification models on the transliterated data from other languages, and use the trained classification models to classify words/sentences of the transliterated test set of the target language. Thus we build our sentiment classification model for a language without using any training data of that language or involving any translation. As shown in the later sections, this method yields in comparable results given by state-of-the-art models like BERT in text classification in scarce-resource scenarios where very little or no training data is available.

3.1 Transliteration - WX notation

We use WX notation (Gupta et al., 2010) – a transliteration scheme for representing Indian languages in ASCII. In this transliteration scheme, every consonant and vowel has a single mapping into Roman. Hence it is a prefix code, advantageous from a computation point of view. In the WX notation, typically lower case letters are used for unaspirated consonants and short vowels while capital case letters are used for aspirated consonants and long vowels. While the retroflexed voiceless and voiced consonants are mapped to 't', 'T', 'd', and 'D', the dentals are mapped to 'w', 'W', 'x', and 'X'. Hence the name of the scheme, WX, refers to the idiosyncratic mapping.

3.2 Word and Sentence Representation

One question that naturally arises is how to represent the words and sentences and what embedding should be used. Our experiments aim to create sentiment lexicon (word level) and classify text (sentence level). Because of this, combined with the fact that there is no corpus in such meta language, the natural choice is using character embedding. Character embedding also provides the benefit that we can have a numeric representation of the new words that get coined to the language. Following dos Santos and Gatti (2014), given a word W composed of M characters $\{c_1, c_2, .r.., c_M\}$, we first transform each character c_m into a character embedding r_m^{chr} . Character embeddings are encoded by column vectors in the embedding $W^{chr} \in \mathrm{R}^{d^{chr} \times |V^{chr}|}$. Given a character c, its embedding r^{chr} is obtained by the matrix-vector product $r^{chr} = W^{chr} v^c$, where v^c is a vector of size $|V^{chr}|$, which has value 1 at index c and zero in all other positions. Thus, W is represented by the sequence of character embeddings $\{r_1^{chr}, r_2^{chr}, r_3^{chr}, \ldots r_M^{chr}\}$.

After transliteration, we label encode each character of this metalanguage and represent words as a vector of these labels (c.f. Figure 1). Each character is one-hot encoded, with the length of the vector set to the number of unique characters in this language. Each of

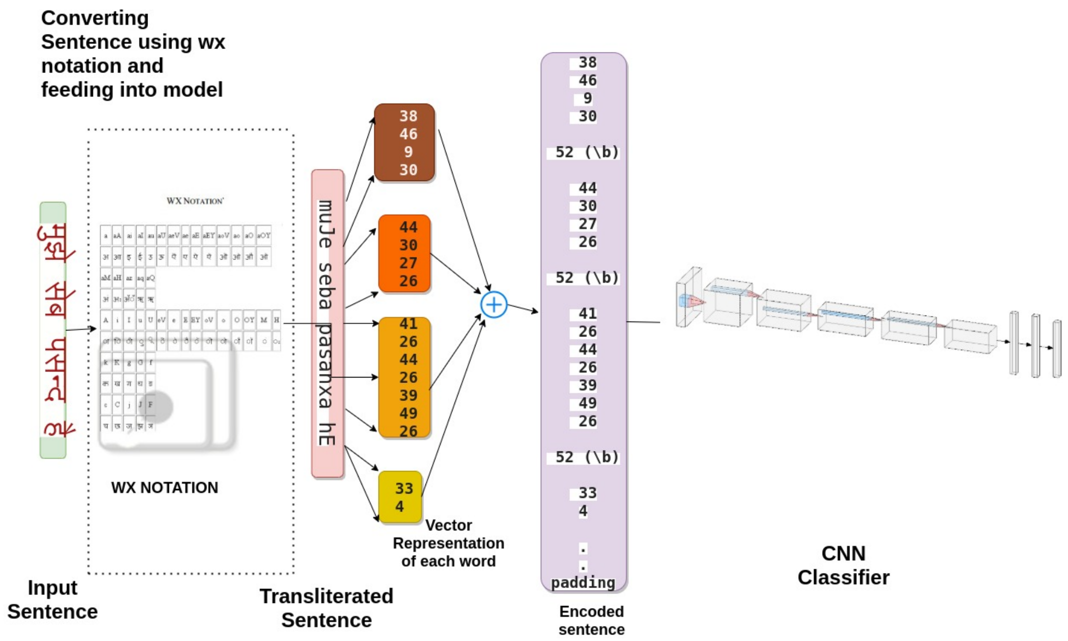

Figure 1: Encoding sentence

these vectors is used as the initial embedding weights for each character. For sentences, the representation differs slightly. Sentences are also represented as a vector of characters, with the difference that an extra character is inserted into the vector which represents space. The embedding weight of this character is initialized as 0, representing a NULL vector. We carried out several experiments to test and compare how the method works under different scenarios, and how to simulate the various situations where this method can be of use, the details of which are given in Section 5.

4 DataSet

We used the SentiWordNet[1] for Indian Languages (Das and Bandyopadhyay, 2010a; Das and Gambäck, 2012; Das and Bandyopadhyay, 2011) for our experiments. This dataset contains sentiment polarity of words in four Indian languages – Bengali (BN), Hindi (HN), Tamil (TA) and Telugu (TE). For each language, the dataset contains four different files LC_POS[2], LC_NEG, and LC_NEU and LC_AMBI, listing the set of positive, negative, neutral, and ambiguous words, respectively. LC_AMBI was not used in our experiments on word-level classification due to the negligible presence of ambiguous words compared to the rest of the classes. Each word in this dataset is marked with POS category[3]. Table 1 presents the statistics of the dataset.

For the actual sentiment analysis task, we used the Aspect Sentiment dataset ($Review_{AH}$), Review Sentiment dataset ($Review_{SH}$) and Movie Review Sentiment dataset ($Movie_H$)[4] (Akhtar et al., 2016). These 3 Hindi datasets are annotated with 4 sentiment classes – positive (POS), negative (NEG), neutral (NEU), and conflict (CON). Table 2 presents the statistics of these datasets.

Language	POS	NEG	NEU	Total
Bengali	1,779	3,714	359	5,852
Hindi	2,313	2,337	371	5,801
Telugu	2,136	4,076	359	6,571
Tamil	2,225	4,447	361	7,033

Table 1: Statistics of the SentiWordNet for Indian Languages

[1] https://amitavadas.com/sentiwordnet.php

[2] LC is the language code – BN: Bengali, HN: Hindi, TA: Tamil, TE: Telugu.

[3] a: Adjective, n: Noun, r: Adverb, v: Verb, u: unknown

[4] www.iitp.ac.in/~ai-nlp-ml/resources.html

Dataset	POS	NEG	NEU	CON	Overall
$Review_{SH}$	2,290	712	2,226	189	5,417
$Movie_{H}$	823	530	598	201	2,152
$Review_{AH}$	1,986	569	1,914	40	4,509

Table 2: Statistics of the Hindi Sentiment Analysis Datasets

5 Experimental Setup

For the classification task, we used various models – RNN, CNN, LSTM, GRU, Self Attention Mechanism (Letarte et al., 2018) and a combination thereof. Embedding length in each model is set to the number of unique characters. For the metalanguage experiments, since the WX format converts characters to uppercase and lowercase English characters, the embedding size is set to 52. Each convolution layer has embedding dimension=1, number of input channel=embedding length, kernel size=1, and number of output feature =3. For LSTM and GRU, the embedding dimension is equal to the input size which in turn is equal to the embedding length, number of hidden states=512, and a dropout rate of 0.01. Each model ended with a dense layer with a softmax activation layer for classification. Many of these layers were combined one on top of the other for sentiment classification of an individual dataset. It was tested with the following learning rates: 0.001,0.005,0.05,0.5 and the best results have been reported. Since the data suffers from class imbalance, proportionate samples were taken from each class. For stacked layers, the hyperparameters were tweaked accordingly to adjust input and output dimensions. For the metalanguage experiments, Adam optimizer was used with a learning rate of 0.001 and sparse categorical cross-entropy as the loss function.

6 Experiments and Results

We conducted experiments in 2 directions – (i) inducing sentiment polarity of words, and (ii) sentence-level sentiment analysis and aspect analysis.

6.1 Sentiment Polarity Prediction of Words

We carried out two sets of experiments for inducing sentiment polarity in words – (a) using training data from the target language, and (b) without using training data from the target language.

We first applied word sentiment classification to individual datasets by using character embedding and different deep learning-based models, namely CNN, LSTM, GRU, Self Attention, and a combination of them. Table 3 presents the results of these experiments.

Different models (along with different learning rates) seem to work better for different languages. For the monolingual setup, GRU, CNN, BiGRU and Self Attention produced the best performance for Bengali, Hindi, Tamil, and Telugu, respectively. Hindi proves to be a challenging language for this task. The best accuracy obtained for Hindi is 51.25 while for the rest 3 languages the accuracy varies in the range [60, 65]. If all the languages and models are considered, CNN with LSTM or GRU layer performs consistently better than the other models.

Following our proposed methodology, for sentiment polarity induction of words in a new language (e.g., TE), we converted both the target language (i.e., TE) and other language (i.e., HN, BN, TA) datasets to a common metalanguage using the wx notation. Then we combined all the other language (i.e., HN, BN, TA) datasets to form the training set and trained our classification models on this dataset. The trained models were then used to predict the sentiment polarity for the target (i.e., TE) dataset. Thus in this experiment, we simulated the scenario where we used no training data of the target language itself, instead used sentiment annotated datasets from other languages for training.

For the cross-language experiments, Self-Attention, CNN+GRU, LSTM, and CNN produced the best results on BN, HN, TA, and TE, respectively. Interestingly, HN proves to be a challenging language for the cross-lingual setup as well.

Table 5 summarizes the best results for each language for both monolingual and cross-

Architecture	Accuracy(%)							
	Monolingual				Cross-Lingual			
	BN	HN	TA	TE	BN	HN	TA	TE
LSTM	64.10	49.17	63.81	61.61	56.53	47.04	**63.76**	60.03
GRU	**65.00**	48.25	62.90	62.16	56.28	47.64	62.80	61.03
BiLSTM	63.10	48.50	62.86	62.27	54.69	49.78	63.08	61.97
BiGRU	64.70	46.75	**64.09**	60.83	56.45	49.60	58.14	57.80
CNN	63.70	**51.25**	63.86	62.38	57.11	49.95	61.80	**62.31**
CNN+LSTM	64.30	50.00	62.86	62.50	56.25	49.75	60.86	61.53
CNN+GRU	64.50	49.00	64.00	62.11	56.33	**50.22**	60.52	60.81
Self Attention	62.70	47.75	62.86	**62.83**	**63.41**	46.96	63.20	62.03

Table 3: Results of Word Sentiment Polarity Identification using Monolingual and Cross-lingual Frameworks

Architecture	Accuracy(%)					
	Using training data			Without Using training data		
	$Review_{SH}$	$Movie_H$	$Review_{AH}$	$Review_{SH}$	$Movie_H$	$Review_{AH}$
Akhtar et al. (2016)	57.34	44.88	65.96	-	-	-
m-BERT	62.69	51.04	59.96	-	-	-
CNN	**61.34**	**46.51**	**68.63**	42.07	**38.38**	**43.88**
LSTM	55.89	42.12	63.30	**42.27**	37.19	42.06
RNN	52.98	42.89	60.86	41.86	33.90	41.48
CNN+LSTM	57.28	43.41	64.04	39.49	36.71	40.11
CNN+GRU	54.28	43.93	64.75	31.47	37.92	43.56
BiGRU	54.50	40.05	62.08	39.36	33.13	39.28
GRU	55.01	40.83	62.64	42.20	34.22	40.66
Self-Attention	53.90	40.13	60.31	37.65	34.72	43.15

Table 4: Results of Sentence-Level Sentiment Analysis and Aspect Analysis

lingual frameworks. It can be observed from Table 5 that the results obtained in the cross-lingual framework are typically lower in accuracy than the results obtained with the monolingual framework, which is quite expected, however, the scores are not very far away. In the absence of any training data for the target language, the results of the cross-lingual experiments can be considered significant.

Train	**Test**	**Architecture**	**Accuracy**
BN	BN	GRU	65.00
TE+HN+TA		Self Attention	63.41
HN	HN	CNN	51.25
BN+HN+TA		CNN+GRU	50.22
TA	TA	BiGRU	64.09
BN+HN+TE		LSTM/GRU	63.76
TE	TE	Self Attention	62.83
BN+HN+TA		CNN	62.31

Table 5: Best results on each testset for the monolingual and metalingual frameworks

6.2 Sentiment Analysis of Sentences and Aspect Analysis

We used the $Review_{SH}$ and $Movie_H$ Hindi datasets (Akhtar et al., 2016) for sentence level sentiment classification and $Review_{AH}$ for aspect term sentiment analysis. Sentences from all these 3 datasets were transliterated from Hindi to the metalanguage with the wx notation. We used the Bengali, Tamil, and Telugu datasets of SentiWordNet (Das and Bandyopadhyay, 2010b) to train the character embeddings which are subsequently used in the classification models. Like the word level sentiment polarity prediction (cf. Section 6.1), we considered two scenarios to classify the sentences – (a) using the sentence-level training data, and (b) without using the sentence-level training data. In both the techniques the language in which the dataset is originally built (Hindi in our case) is not directly used for training the model. Instead, the sentences are transliterated to the wx notation and encoded using the technique specified earlier.

80% of the dataset was used as training data and 20% was treated as the testset. Evaluation results are reported in Table 4 under the column “Using training data”. We received the best accuracy of 61.34, 46.51, and 68.63 on the $Review_{SH}$, $Movie_H$ and $Review_{AH}$ datasets, respectively, which are significantly better than the results reported in (Akhtar et al., 2016). CNN produced the best results across all the datasets. We also fine-tuned BERT on the datasets using multilingual-BERT(m-BERT) as an encoder followed by a simple classification header. Although BERT outperforms in most cases, our scores are still comparable to those obtained with BERT.

For the case where sentence level training data is not used, we did not use any portion of the $Review_{SH}$, $Movie_H$ and $Review_{AH}$ datasets for training. Instead, we used transfer learning where the model that was trained to determine the embedding weights of each character and classify words is used to classify the sentences. It is to be noted that we did not even use the Hindi dataset of SentiWordNet to train the character embeddings for this experiment. The evaluation results are shown in Table 4 under the column “Without using training data”. As expected, the obtained results are much lower than the corresponding results reported under column "Using training data" in Table 4. However, the results suggest that in a resource-constrained scenario where there is no training data available, this method can act as an effective way of classification.

7 Analysis of Results

Results suggest that our proposed method performs reasonably well for different tasks and languages. The traditional CNN architectures captured structural features very well, which is evident from the fact that sentiment embedded vectors when incorporated in training produced state-of-the-art results on most of the datasets. We realize that using CNN will give us two main advantages: (i) learn hidden semantics from a metalanguage, and (ii) handling limited coverage of lexical resources. Even when training data is not used, we achieved promising results which proves the effectiveness of the method in a resource-constrained scenario. An interesting observation was found in the $Review_{AH}$ dataset, where the model gave different results based on the convolution window size. For example, the models performed better when the sentence comprised of the aspect term and four words from either side of the aspect term and used for sentiment classification than the situation when 2 words from either side were considered for training. This can be attributed to the fact that the information from the distant part of the sentences is sometimes attributed to the overall sentiment polarity of the aspect terms. However, the situation reversed when no training data was used, where sentences with 2 words from either side provided better classification results in comparison to sentences with 4 words from either side of aspect terms. This is because we used word-level classification models to classify these sentences, and hence these models captured the local features of the aspect terms more efficiently and gave better accuracy.

8 Conclusions

In this paper, we introduced the novel task of inducing the sentiment polarity of words using character embedding-based deep learning models. We extended the task to inducing the sentiment polarity of words in a new language having no training data. We carried out experiments with 4 Indian languages and obtained encouraging results. The cross-lingual approach proved to be an effective method in a resource-constrained scenario. The same idea was followed to perform sentiment analysis and aspect analysis in Hindi without using any training data in Hindi. While using training data, our method outperformed the previous state-of-the-art in sentiment analysis and aspect analysis in Hindi.

References

Mohamed Abdalla and Graeme Hirst. 2017. Cross-lingual sentiment analysis without (good) translation. *arXiv preprint arXiv:1707.01626*.

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 482–493.

AR Balamurali, Aditya Joshi, and Pushpak Bhattacharyya. 2012. Cross-lingual sentiment analysis for indian languages using linked wordnets. In *Proceedings of COLING 2012: Posters*, pages 73–82.

Grzegorz Chrupała. 2013. Text segmentation with character-level text embeddings. *arXiv preprint arXiv:1309.4628*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Amitava Das and Sivaji Bandyopadhyay. 2010a. Sentiwordnet for indian languages. In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pages 56–63.

Amitava Das and Sivaji Bandyopadhyay. 2010b. SentiWordNet for Indian languages. In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pages 56–63, Beijing, China. Coling 2010 Organizing Committee.

Amitava Das and Sivaji Bandyopadhyay. 2011. Dr sentiment knows everything! In *Proceedings of the ACL-HLT 2011 System Demonstrations*, pages 50–55, Portland, Oregon. Association for Computational Linguistics.

Amitava Das and Björn Gambäck. 2012. Sentimantics: Conceptual spaces for lexical sentiment polarity representation with contextuality. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 38–46, Jeju, Korea. Association for Computational Linguistics.

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

Cícero dos Santos and Maíra Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Imane Guellil, Ahsan Adeel, Faical Azouaou, Fodil Benali, Ala-eddine Hachani, and Amir Hussain. 2018. Arabizi sentiment analysis based on transliteration and automatic corpus annotation. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 335–341, Brussels, Belgium. Association for Computational Linguistics.

Rohit Gupta, Pulkit Goyal, and Sapan Diwakar. 2010. Transliteration among indian languages using wx notation. In *KONVENS*.

Sarthak Jain and Shashank Batra. 2015. Cross lingual sentiment analysis using modified brae. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 159–168.

Gyuwan Kim. 2019. Subword language model for query auto-completion. *arXiv preprint arXiv:1909.00599*.

Divya Kumari, Asif Ekbal, Rejwanul Haque, Pushpak Bhattacharyya, and Andy Way. 2021. Reinforced NMT for sentiment and content preservation in low-resource scenario. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 20(4):70:1–70:27.

Gaël Letarte, Frédérik Paradis, Philippe Giguère, and François Laviolette. 2018. Importance of self-attention for sentiment analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 267–275, Brussels, Belgium. Association for Computational Linguistics.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pintu Lohar, Haithem Afli, and Andy Way. 2017. Maintaining sentiment polarity in translation of user-generated content. *The Prague Bulletin of Mathematical Linguistics*, 108(1):73.

Pintu Lohar, Haithem Afli, and Andy Way. 2018. Balancing translation quality and sentiment preservation (non-archival extended abstract). In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 81–88.

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. 2012. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 8(67).

Santanu Pal, Braja Gopal Patra, Dipankar Das, Sudip Kumar Naskar, Sivaji Bandyopadhyay, and Josef van Genabith. 2014. How sentiment analysis can help machine translation. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 89–94.

Mohammad Sadegh Rasooli and Michael Collins. 2017. Cross-lingual syntactic transfer with limited resources. *Transactions of the Association for Computational Linguistics*, 5:279–293.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 111–121.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Cross-lingual sentiment classification with bilingual document representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1412.