A Hindi Sentiment Analysis Corpus from Amazon Reviews

Udrasht Pal, Nikhil Khemchandani, Radhika Mamidi
IIIT Hyderabad, India
{udrasht.pal, nikhil.khemchandani}@students.iiit.ac.in
{radhika.mamidi}@iiit.ac.in

1 Abstract

The analysis of sentiment is crucial for evaluating online content in various languages, serving purposes like content moderation and opinion mining. Despite the abundance of resources for sentiment analysis in numerous Indian languages, there is a notable absence of large-scale, freely accessible datasets for Hindi. Our paper introduces and delineates the Hindi Sentiment Analysis Corpus (HSAC), derived from Amazon English reviews and translating them to Hindi language. We meticulously outline our procedures for gathering and annotating data and perform comprehensive experiments on our corpus to establish robust benchmarks for future research utilizing our dataset.

2 Introduction

Sentiment Analysis, an essential task in natural language processing, involves determining the sentiment or polarity (positive, negative, neutral) of text (Pang et al., 2008). With the exponential growth in internet access and social media usage across languages like English, Spanish, French,etc, it has garnered significant attention. However, there has been relatively little exploration of Amazon reviews in Indian languages such as Hindi.

Hindi holds a prominent position in Northern India, boasting over 260 million native speakers and a significant presence in states like Uttar Pradesh, Bihar, and Madhya Pradesh. As the official language of India, it bears immense cultural and linguistic significance. Despite the substantial engagement of its online community on social media platforms and its widespread representation in mainstream media, there is a conspicuous absence of comprehensive, publicly accessible resources for sentiment analysis in Hindi.

Therefore, we introduce a new dataset for Hindi language which is Hindi Sentiment Analysis Corpus (HSAC) tailored for monolingual sentiment classification. Sourced from Amazon reviews along with their ratings, we elucidate our meticulous annotation process. Furthermore, we conduct extensive experiments on the dataset, employing both feature-based and deep learning architectures, to establish a dependable baseline for HSAC. We also compare the performances of various model architectures. The dataset is readily available on GitHub.

2 Related Work:

Their are so many work done in hindi sentimental analysis on twitter data, like (Uncovering Political Hate Speech During Indian Election Campaign) in this they created india state election hate and non hate data and predicted the sentiment analysis, they took data from twitter (2022 tweets) of states elections and then applied machine learning algorithm and found the accuracy.

Another related work “A Gujarati Sentiment Analysis Corpus from Twitter” they created data from twitter in english and converted it into gujarati and applied machine learning algorithms.

Apart from that “Sentiment analysis on large scale Amazon product reviews” they create the data in english and apply machine learning algorithms to it.

There is no specific work done on hindi amazon reviews in our knowledge, so our work is unique and that's why we have done this.

3 Dataset Creation:

Dataset link:
https://github.com/Nikhil5555/A-Hindi-Sentiment-Analysis-Corpus-from-Amazon-Reviews/tree/main/data

3.1 Collection:

We sourced our dataset from Amazon, initially focusing on collecting general reviews rather than targeting any specific language. The wide array of electronic products available on Amazon provided us with abundant data for analysis.

Initially, we meticulously gathered reviews from various electronic product categories, including phones, laptops, earphones, monitors, speakers, chargers, power banks, electronic watches, trimmers, electric fans, and electric kettles. Through manual effort, we collected approximately 710 product review links. Within each category, we ensured representation from various manufacturers and brands, aiming for a comprehensive and diverse dataset. Typically, each link provided up to 10 reviews, enriching the breadth and depth of our dataset.

Using the BeautifulSoup Python library, we performed web scraping to extract review data from the collected links. This initial process resulted in a dataset comprising 7,054 reviews. However, to ensure the quality and relevance of the data, we implemented data cleaning procedures. Firstly, we filtered out reviews that were not in English, leading to the removal of approximately 2,000 reviews. Additionally, we discarded reviews with titles and content exceeding 5000 characters, which facilitated the subsequent translation processes. This filtering reduced our dataset to approximately 5,000 reviews.

The reduction from 7,054 to 5,000 reviews attributed to various factors:

- Some reviews lacked content or only contained emojis, making them unsuitable for analysis.
- A portion of the reviews were in languages other than English, such as Japanese, Hindi, Hindi-English, and Chinese, which were not relevant to our study.
- Inconsistencies in language usage and content quality further necessitated the removal of certain reviews to maintain data integrity and accuracy.

Subsequently, we utilized Google Translate to convert the English titles and content into Hindi. This translation process produced Hindi versions of the titles and content, resulting in approximately 4,411 translated reviews for our dataset.

The complete process we followed is described below:

- Manual collection of product review links from Amazon across various electronic categories and manufacturers.
- Web scraping of review data using the BeautifulSoup library to extract essential information.
- Data cleaning procedures to filter out non-English reviews, reviews lacking content, and reviews with excessive character lengths.
- Translation of English reviews to Hindi using Google Translate to create a bilingual dataset for analysis.

3.2 Annotation Based on Rating:

In our annotation process, we categorize reviews based on the product rating provided by customers. This approach allows for a straightforward classification of sentiment into positive, negative, and neutral categories, providing valuable insights into customer perceptions and preferences.

3.2.1 Reasons for Choosing the Rating Column:

- **Objective Measure**: Ratings offer a quantifiable measure of customer satisfaction, serving as an objective criterion for sentiment classification.
- **Ease of Interpretation**: Ratings are intuitive and easily understandable, facilitating straightforward annotation and analysis.
- **Widespread Usage:** Ratings are extensively utilized across e-commerce platforms like Amazon, making them readily available and familiar for annotation and analysis.
- **Trustworthiness:** Higher ratings (4 and 5) often signify high levels of customer satisfaction and trust in the product, making them reliable indicators of positive sentiment.
- **Consistency in Sentiment:** Reviews with higher ratings commonly exhibit consistent positive language and expressions, further supporting their classification as "Positive."

3.2.2 Annotation Guidelines:

- **"Positive":** Reviews with ratings between 4 and 5, indicating high customer satisfaction and positive sentiment towards the product.
- **"Negative":** Reviews with ratings 1 and 2, signifying low customer satisfaction and negative sentiment towards the product.
- **"Neutral":** Reviews with a rating of 3, indicating neither overtly positive nor negative sentiment towards the product.

3.3.3 Enhanced Insights:

By annotating reviews based on ratings, we gain deeper insights into customer sentiment and preferences. This method allows for a comprehensive analysis of customer feedback, enabling businesses to identify areas for improvement and capitalize on strengths. Furthermore, it provides a standardized framework for sentiment classification, enhancing the consistency and reliability of analysis outcomes.

3.4 Statistics:

Our final dataset contains a total of 4,411 reviews. We divide the dataset into training, and test sets in a 80:20 ratio, respectively. Within the complete dataset, the `negative` class has the highest representation, comprising about 44.5% of the total dataset, followed by `Positive` at 36.05% and finally `neutral` at 19.3%.

Additional details about the class distribution are reported in Table 2.

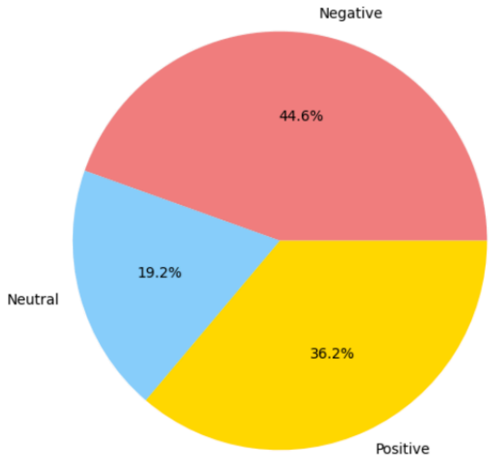

Figure 1:Split-wise Class Distribution of Dataset

Split	Positive	Neutral	Negative	Total Count
Train	1276	680	1571	3527
Test	320	170	394	884
Total	1596	850	1965	4411

Table 2: Split-wise Class Distribution of Dataset

S.NO	title	content	title_hindi	content_hindi	Review_type
1	Worst camera experience . Better to go for MI	Overall experience is very bad since I received it.1 It has connectivity issue …	सबसे ख़राब कैमरा अनुभव. एमआई के लिए जाना बेहतर है	जब से मुझे यह प्राप्त हुआ है, कुल मिलाकर अनुभव बहुत खराब है।1. इसमें वाईफाई के साथ …	Negative
2	good mobile in this price range	The phone is good, but the camera quality is not that good…	इस मूल्य सीमा में अच्छा मोबाइल	फोन अच्छा है, लेकिन कैमरे की गुणवत्ता उतनी अच्छी नहीं है …	Neutral
3	Without a doubt the best tablet option available under Rs 30,000.	I originally bought it for productivity purposes. No use for gaming at all…	बिना किसी संदेह के 30,000 से कम में उपलब्ध सर्वोत्तम टैबलेट विकल्प।	मैंने इसे मूल रूप से उत्पादकता उद्देश्यों के लिए खरीदा था। गेमिंग के लिए बिल्कुल भी उपयोग नहीं …	Positive

Table 3: Some samples from the HSAC dataset

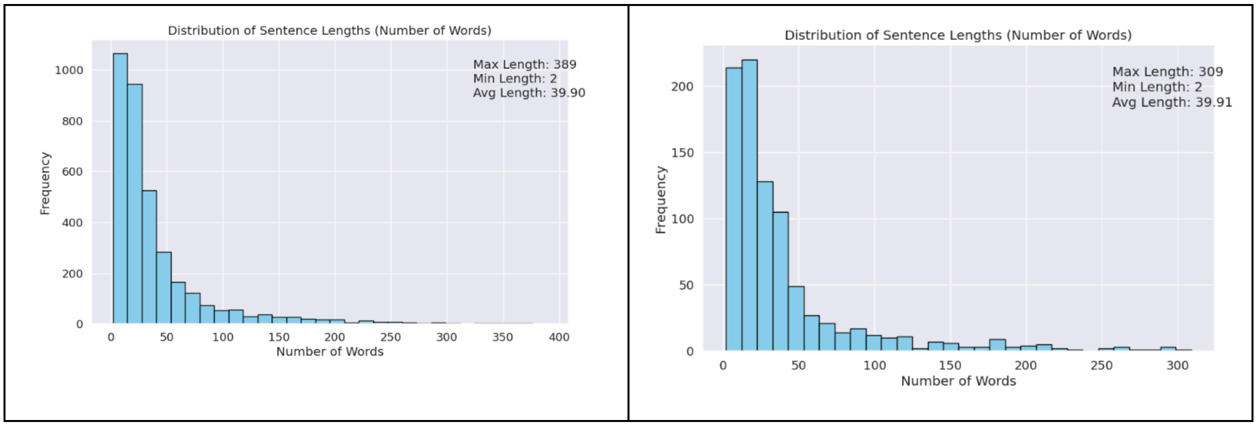

Figure 2: Left figure is for train and Right is for test

4 Experiments:

Code link:

https://github.com/Nikhil5555/A-Hindi-Sentiment-Analysis-Corpus-from-Amazon-Reviews/tree/main/code

We train using six different types of models to test how different models perform on our dataset and to set baselines for it. Four models are feature vector-based models, which we train on two different variants based on different sets of features - Bag-of-Words and TF-IDF, which are SVM, Logistic Regression, Decision Tree and Naive Bayes. And the last two models are based on the Deep Neural Network Model i.e. the MLP and LSTM model.

4.1 Feature Vector Models

We train four classifiers - Naive Bayes, Logistic Regression, Support Vector Machines, and Decision Tree, each on two different feature vectors - Bag-of-Words and TF-IDF for a total of 8 models.

Bag-of-Words (BoW) or Countvectorizer represents a document (in this case, a review) as a vector of the counts of each word present in the document.Even though it ignores word order, bag-of-words features can still be useful as feature vectors for tasks such as text classification (McCallum and Nigam, 2001).

TF-IDF (Term Frequency - Inverse Document Frequency) (Spärck Jones, 1972) is a method to represent documents that factors in the relative frequency of a word across documents by calculating a score based on two parameters - term frequency,which is the frequency of a term in the current document, and inverse document frequency - which is based on the frequency of the term across all documents.

The models we train for each of these are:

Naive Bayes Classifier -The Naive Bayes classifier is a straightforward model that estimates the probability of each label by assuming that input features are conditionally independent. This approach has demonstrated effectiveness, particularly in text classification tasks (McCallum and Nigam, 2001).

Logistic Regression - Logistic regression (Cox, 1958) is a classification algorithm that models the probability of an input feature belonging to specific classes using a logistic function. We train a logistic regression classifier over 1000 epochs or until convergence, employing a one-vs-all strategy.

Support Vector Machine - A support vector machine (Cortes and Vapnik, 1995) is a classifier that seeks to identify the hyperplane that best separates the training data according to their labels. This method is also trained using a one-vs-all approach.

Decision Trees - Decision trees(Ross Quinlan,1986) are a type of classifier which construct a tree-like structure where each internal node represents a decision based on a feature, and each leaf node represents a class label. Decision trees recursively split the training data based on features to maximize information gain or minimize impurity, aiming to create homogeneous subsets. We used gini impurity.

Multi-Layer Perceptron The Multi-Layer Perceptron (MLP) is a foundational concept in neural networks, (developed by Rosenblatt in 1958 and refined by Rumelhart et al). in 1986. Our model employs an MLP with 100 and 300-dimensional hidden layers, using ReLU activation functions. Trained over multiple epochs, it adjusts parameters to minimize errors, aiming to capture complex data patterns for insightful predictions.

Long Short-Term Memory (LSTM) - ELMo (Embeddings from Language Models), combined with LSTM (Long Short-Term Memory), forms a potent architecture designed to address the vanishing gradient problem inherent in traditional RNNs while excelling at capturing long-range dependencies in sequential data. LSTM units feature memory cells capable of retaining information across time steps, facilitating selective retention or forgetting of data. This combination is particularly effective in tasks like sequence prediction, language modeling, and sentiment analysis.

In our model, we integrate the ELMo method with LSTM for enhanced contextual embeddings. ELMo provides a deep, context-aware representation of words, enriching the input data for the LSTM layer. We set the embedding dimension to 300 to accommodate the richer contextual information provided by ELMo.

Considering a BATCH_SIZE of 1, the model architecture comprises a single LSTM layer with a hidden size of 150 units. Each LSTM unit is activated using the rectified linear unit (ReLU) activation function, facilitating non-linearity in the model.

Key hyperparameters for training include a learning rate of 0.001 and a training duration of 10 epochs. During training, the model learns to extract intricate patterns and dependencies from the input data, leveraging both the power of ELMo embeddings and the memory capabilities of LSTM units.

Bag of Words:

Model	Precision	Recall	Accuracy	Weighted F1	Macro F1
Naive Bayes(unigram)	0.59	0.58	0.66	0.64	0.57
Naive Bayes(bigram)	0.70	0.63	0.73	0.69	0.61
Logistic Regression(unigram)	0.61	0.60	0.68	0.66	0.60
Logistic Regression(bigram)	0.68	**0.65**	**0.73**	**0.71**	**0.65**
Support Vector Machine(unigram)	0.66	0.60	0.71	0.66	0.57
Support Vector Machine(bigram)	**0.81**	0.60	0.71	0.66	0.57
Decision Tree(unigram)	0.60	0.59	0.65	0.65	0.60
Decision Tree(bigram)	0.15	0.33	0.45	0.27	0.21

Table 4: Results when using the Bag of words

TFIDF:

Model	Precision	Recall	Accuracy	Weighted F1	Macro F1
Naive Bayes(unigram)	0.46	0.55	0.68	0.60	0.50
Naive Bayes(bigram)	**0.83**	0.58	0.71	0.64	0.54
Logistic Regression(unigram)	0.64	0.62	0.71	0.68	0.61
Logistic Regression(bigram)	0.77	0.63	**0.74**	0.69	0.61
Support Vector Machine(unigram)	0.73	**0.64**	**0.74**	**0.70**	**0.63**
Support Vector Machine(bigram)	0.81	0.63	**0.74**	0.69	0.61
Decision Tree(unigram)	0.57	0.57	0.64	0.57	0.62
Decision Tree(bigram)	0.15	0.33	0.45	0.27	0.21

Table 5: Results when using the TF-IDF

Neural Networks:

Model	Precision	Recall	Accuracy	Weighted F1	Macro F1
FFNN	0.69	0.70	0.70	0.69	0.64
LSTM+ELMO	0.67	0.70	**0.71**	0.68	0.62

Table 5: Results of Neural networks models

Results:

The Naive Bayes model(Bigram) and SVM(bigram) using TF-IDF features achieves the highest precision(>=0.81) out of all the models trained. Other statistical models like Logistic Regression also achieve reasonably high average precision (>= 0.77) while taking significantly less computational resources and time.

When it comes to neural networks, the FFNN gets things right about 70% of the time, and the LSTM+Elmo does a bit better at 71%. Even though they're not as accurate as some other models, they're still quite good at understanding complex patterns in the data.

Conclusion:

In this study, we explore the Hindi Sentiment Analysis Corpus (HSAC), comprising over 4411 reviews. To our knowledge, it stands as the first substantial publicly accessible corpus sourced from Amazon reviews tailored for sentiment analysis in Hindi. Additionally, we introduce our annotation schema and conduct extensive experimentation to establish baseline performance metrics for this novel dataset. Notably, we curate the dataset ourselves and apply various machine learning techniques to analyze it comprehensively.

When comparing the accuracy of neural networks like FFNN and LSTM+Elmo to models like Naive Bayes, SVM, and Logistic Regression, it's important to consider several factors. One key reason for the lower accuracy of neural networks in this case could be the size of the dataset. With only 4411 reviews available for training, the neural networks might not have had enough diverse examples to learn from compared to the larger datasets used for models like Logistic Regression.

Additionally, the complexity of neural networks introduces a higher risk of overfitting, especially when the dataset is small. This means that the neural networks might be learning to memorize the training data rather than generalize well to unseen data, leading to slightly lower accuracy.

Despite these challenges, neural networks, including FFNN and LSTM+Elmo, still demonstrate impressive capabilities in capturing intricate patterns within the data. Their performance, though slightly lower than TF-IDF-based models like Logistic Regression, showcases their potential for understanding complex data structures, making them valuable tools in various applications.

References:

For scrip Amazon review https://oxylabs.io/blog/how-to-scrape-amazon-reviews

Twitter Sentiment Analysis
https://www.researchgate.net/publication/352780855_Twitter_Sentiment_Analysis_using_Deep_Learning

GSAC: A Gujarati Sentiment Analysis Corpus from Twitter
M Gokani, R Mamidi - Proceedings of the 13th Workshop on Computational …, 2023

Uncovering Political Hate Speech During Indian Election Campaign:
A New Low-Resource Dataset and Baselines

Andrew McCallum and Kamal Nigam. 2001. A comparison of event models for naive bayes text classification. Work Learn Text Categ, 752.

David R Cox. 1958. The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2):215–232.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, 28(1):11–21

Sven Buechel, Susanna Rücker, and Udo Hahn. 2020. Learning and evaluating emotion lexicons for 91 languages. CoRR, abs/2005.05672.

Bharathi Raja Chakravarthi, KP Soman, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Kingston Pal Thamburaj, John P McCrae, et al. 2021. Dravidianmultimodality: A dataset for multi-modal sentiment analysis in tamil and malayalam. arXiv preprint arXiv:2106.04853.

Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 383–389, Baltimore, Maryland. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Supportvector networks. Machine learning, 20(3):273–297.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. arXiv preprint arXiv:2211.11418.

Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. 2021b. L3cubemahasent: A marathi tweet-based sentiment analysis dataset. CoRR, abs/2103.11408.

Sentiment analysis on large scale Amazon product reviews Conference: 2018 IEEE International Conference on Innovative Research and Development (ICIRD)
At: Bangkok, Thailand

Evaluation of Deep Learning Models for Hostility
Detection in Hindi Text arXiv:2101.04144v4 [cs.CL] 7 Apr 2021

Sentiment Analysis of Hindi Review based on Negation and Discourse Relation International Joint Conference on Natural Language Processing, pages 45–50,
Nagoya, Japan, 14-18 October 2013.