Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute InformingScience.org

IJIKM.org

Volume 20, 2025

NOVEL FINE-TUNED BIDIRECTIONAL GRU AND FASTTEXT EMBEDDINGS FOR LOCATION-BASED SENTIMENT ANALYSIS AND PREDICTIONS

Akshatha Shetty	Dept of MCA, St. Agnes College (Autonomous), Mangalore, India	akshathagp12@gmail.com
Manjaiah D H	Dept of Comp Science, Mangalore University, Mangalore, India	drmdhmu@gmail.com
Praveena Kumari M. K.*	Dept of MCA, NMAM Institute of Technology NITTE (Deemed to be University), Nitte, India	narayan.praveena@gmail.com

Corresponding author

ABSTRACT

Aim/Purpose	The need for this paper stems from the challenge of efficiently analyzing large volumes of customer reviews in the hotel industry, which is growing and complex due to the widespread use of digital platforms. With consumers increasingly sharing feedback across various social media applications, manual processing becomes impractical, necessitating machine learning algorithms for accurate sentiment analysis and prediction.
Background	This paper addresses the problem by applying machine learning algorithms, specifically fine-tuned bidirectional GRU with FastText embedding, to perform location-based sentiment analysis and prediction of customer reviews in the hotel industry, providing an efficient solution to process large-scale unstructured data and extract valuable insights for improving customer experience.
Methodology	This study employs machine learning techniques, including Random Forest (RF), Support Vector Machine (SVM), Bidirectional Long-Short Term Memory (BiLSTM), and Bidirectional Gated Recurrent Unit (GRU), to perform sentiment analysis on customer reviews in the hotel industry. The paper focuses on using a fine-tuned bidirectional GRU model with FastText embedding for location-based sentiment analysis and prediction. The research sample consists of a large dataset of customer reviews, which are processed and analyzed to predict

Accepting Editor Geoffrey Z. Liu | Received: February 26, 2025 | Revised: April 24, April 25, April 30, 2025 | Accepted: May 1, 2025.
Cite as: Shetty, A., Manjaiah, D H, & Praveena Kumari, M. K. (2025). Novel fine-tuned bidirectional GRU and FastText embeddings for location-based sentiment analysis and predictions. *Interdisciplinary Journal of Information, Knowledge, and Management, 20,* Article 15. https://doi.org/10.28945/5515

	sentiment and evaluate model performance, achieving an overall score of 84.31%.
Contribution	This paper contributes to the body of knowledge by demonstrating the effective application of advanced machine learning algorithms, particularly a fine-tuned bidirectional GRU with FastText embedding, for sentiment analysis in the hotel industry. It offers a unique approach to location-based sentiment analysis, allowing for a deeper understanding of regional variations in customer perceptions. The study also provides insights into the comparative effectiveness of different machine learning models for sentiment classification and prediction by comparing multiple algorithms. This work enhances the existing methodologies for processing large-scale, unstructured data and highlights the potential of sentiment analysis to drive data-informed strategies in customer-centric industries.
Findings	The study found that the fine-tuned bidirectional GRU model with FastText embedding achieved an accuracy of 84.31%, outperforming other algorithms in sentiment classification. It highlighted the effectiveness of location-based sentiment analysis for understanding regional variations in customer perceptions. The approach also demonstrated strong predictive capabilities, aiding data-driven decision-making in the hospitality sector.
Impact on Society	The findings of this paper have significant implications for industries, particularly in the hospitality sector, as they provide an effective method for analysing customer sentiment at a granular level. The ability to assess customer opinions through sentiment analysis can help businesses improve customer service, tailor their offerings to meet customer needs, and enhance the overall customer experience. Additionally, location-based sentiment analysis can enable businesses to understand regional differences, allowing for more personalized marketing strategies and better resource allocation.
Future Research	Future research could focus on improving the accuracy and efficiency of sentiment analysis models by incorporating more advanced deep learning techniques and larger, more diverse datasets. Exploring the application of sentiment analysis in other sectors, such as healthcare, education, and finance, could provide broader insights into customer perception. Additionally, integrating sentiment analysis with other data sources, such as social media and customer feedback, may lead to more comprehensive and real-time prediction models. Further directions include multilingual sentiment analysis, the use of transformer-based models, and deployment in resource-constrained environments.

Keywords sentiment analysis, bidirectional GRU, fast-text embeddings, location

INTRODUCTION

An important area to investigate is Sentiment Analysis (SA), i.e., opinions, views, and sentiments in Natural Language Processing (NLP). The polarity of the text is examined using SA. This knowledge can help decision-making and provide better illustrations for predicting trends and tackling abnormal situations. Finding the sentiment or polarity in words or texts is the main challenge in SA. The work of Turney (2002) and Pang et al. (2002), using various approaches to recognize the polarity of product reviews and movie reviews, respectively, was one of several later efforts that were less sophisticated and only used a simple polar perspective of sentiment, from positive to negative. Studies should first develop a proper method for analyzing text and further classifying what each segment implies to gain knowledge and understanding from only a string of words. There are various

approaches to accomplish this, most based on classifier understanding. Most methods have issues such as the need for domain-specific datasets, higher classification errors, the need for domain knowledge, time constraints, etc. SA is now a widely used research area with many applications. Dealing with the structural features of the text while assessing its expressed sentiment is one of the major research questions. Word embedding is a method to map high-dimensional, sparse vectors that denote the words into a real-number vector in continuous space, offering a notable reduction in dimensionality, depending on the word co-occurrence in a large collection of text. This representation is known for grouping similar words and showing how they relate, which is implemented to identify analogies (Mikolov, Chen, et al., 2013).

While many approaches have made progress in sentiment analysis, they often fail to generalize well across different linguistic styles or handle informal language, emojis, and abbreviations common in user reviews. Furthermore, regional sentiment variation is an underexplored dimension in existing models. One of the limitations is using a single data source; possible improvement solutions may include data augmentation and an increase in data sources. The small size of the data does not ensure that accurate predictions and bias may increase in such cases. Information on other social media has so far been useful in gathering insightful data from travelers (Serna et al., 2016). Also, exploring more recent and comprehensive data sets should allow them to compare with the latest information. To examine unknown patterns, sentiment scores can be combined with various data types from multiple fields, such as transportation, environment, weather, etc. NLP-based review classification also has the potential to handle multilingual review classification in the long term. Ma et al. (2018) used datasets collected from TripAdvisor and Yelp websites. They have compared deep learning models with another ML approach to explore the effect of user-provided photos on review usefulness. Their results showed that the deep learning model is more useful in predicting reviews compared with other ML models.

Mikolov, Chen, et al. (2013) developed N-grams and Word2Vec embeddings for continuous vector representation. Word2Vec uses the Feedforward network. There are layers for the projection, input, and output. For NLP tasks (Pennington et al., 2014), word representations are performed using Global Vector (GloVe) embedding. It utilizes the features of both local context window approaches and global matrix factorization. Two other word-embedding techniques that are widely used are FastText, proposed by Bojanowski et al. (2017), and ELMo, developed by Peters et al. (2018). In their study on sentiment analysis of the large IMDB movie review dataset, Sivakumar and Rajalakshmi (2019) employed embedding techniques such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec to extract features. These features were then integrated with classifiers such as Stochastic Gradient Descent (SGD), Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), and Decision Tree. Their results proved that Word2Vec and SGD are the best for classifying sentiment problems. Harish et al. (2019) also performed SA on IMDb movie reviews and detected the reviewers' sentiments about a movie using a hybrid feature extraction method. These features are extracted by combining TF or TF-IDF with Lexicon features such as Positive-Negative word count, Connotation. Many researchers are attempting to refine the SA model to detect both positive and negative reviews. The polarity of a sentence's sentiments is significantly influenced by its sentence form. Some aspects of the sentence are more important to determine the sentence polarity when modifiers such as conjunctions, conditionals, and intensives are present. Li et al. (2020) applied average pooling over the output of the word-level attention layer to obtain a fixed-length sentence representation for sentiment classification.

Research Gaps and Significance

Recent developments in deep learning, such as transformer-based architectures like BERT and RoBERTa, have demonstrated strong performance in sentiment analysis tasks. However, these models are often limited by their high resource demands and are rarely applied in location-sensitive domains like the hospitality industry, representing a promising direction for future research. While sentiment analysis has been widely explored, several critical gaps remain in the current literature:

- Limited exploration of location-based sentiment analysis within the hospitality domain.
- Insufficient investigation of Bidirectional GRU architectures combined with modern embedding techniques.
- Lack of comprehensive comparative analyses between traditional machine learning and deep learning approaches for hotel review classification.
- Inadequate preprocessing techniques to handle informal text, emojis, and abbreviations in customer reviews.

This research addresses these gaps by:

- Proposing a novel combination of Bidirectional GRU with FastText embeddings.
- Implementing a comprehensive data purification pipeline tailored to noisy, user-generated content.
- Providing location-specific sentiment analysis for enhanced understanding of regional customer perceptions.
- Conducting extensive comparative analysis against baseline models.

Our proposed system applies machine learning algorithms – specifically, a fine-tuned Bidirectional GRU with FastText embeddings – to perform location-based sentiment analysis and predict customer reviews in the hotel industry. This approach offers an efficient solution for processing large-scale unstructured data and extracting valuable insights to improve customer experience.

The remainder of this paper is structured as follows. The next section presents related research, which is followed by the proposed methodology, the experimental results and analysis, and the key findings and future directions.

LITERATURE REVIEW

This section specifies an overview of the prior study on SA, including approaches for sentiment classification, feature extraction from text, and text mining. Text mining is considered traditional data mining with text features.

Table 1. Summary table of literature

Study	Approach/ method	Dataset	Key findings	Model performance	Key features	Gap/ shortcoming
Yao et al. (2011)	Keyword extraction for sentiment classification	Reviews (polarity)	Keywords categorized reviews into positive or negative.	Not specified	Extracted keywords based on the language model	Focused on keyword extraction, not on embedding-based or hybrid models.
Farisi et al. (2019)	Multinomial Naïve Bayes (NBM) for sentiment analysis	Hotel reviews	Achieved a high F1 score for positive/negative classification.	Avg. F1 score > 91%	NBM classifier with feature extraction post-data preprocessing	Lack of deep learning approaches and potentially less generalizability.
Haripriya et al. (2018)	Hybrid method (NB + SVM, sentiment lexicon, n-grams)	Social media reviews (trending events)	A hybrid approach was evaluated for various combinations to determine model performance.	Varies by model and setup	N-grams (unigram, bigram), sentiment lexicon	Results vary, no clear winner, and lack of focus on deep learning models or word embeddings.

Study	Approach/ method	Dataset	Key findings	Model performance	Key features	Gap/ shortcoming
Afzaal et al. (2019)	Multinomial Naïve Bayes (NBM) for sentiment classification	Restaurant and hotel reviews	NBM achieves high accuracy for restaurant and hotel reviews.	Restaurant: 88.08%, Hotels: 90.53%	NBM classifier, feature extraction via TF-IDF	Limited to NBM; no exploration of neural networks or deep learning models.
Laoh et al. (2019)	SVM with n-grams for sentiment classification	Reviews	SVM with bigrams achieved higher accuracy than unigram-based classifiers.	Unigram: 90%, Bigram: 94%	Unigram and bigram features for SVM classification	Focused on SVM with n-grams, lacks exploration of deep learning or word embeddings.
Mishev et al. (2019)	Word2Vec, Glove, FastText embeddings + ML & deep learning	Financial text	Deep learning with embeddings (Bi-GRU, Attention) outperformed other methods.	Best score with Bi-GRU and Attention	Word2Vec, Glove, FastText embeddings, Bi-GRU, Attention	Focused on financial texts, embeddings not combined with sentiment-specific analysis for broader domains.
Yaqub et al. (2020)	Location-based sentiment analysis	US and UK election data	Location-based sentiment reflects public opinion during elections.	Not specified	Location-based sentiment, political analysis	Lacks exploration of embedding-based sentiment analysis or deep learning for general sentiment classification.
Puh and Bagić Babac (2023)	Various classifiers (NB, SVM, CNN, LSTM, BiLSTM) for reviews	Visitor reviews	Compared multiple classifiers for sentiment classification of reviews.	Varies by model; CNN, LSTM, BiLSTM used	TF-IDF (NB, SVM), Glove (CNN, LSTM, BiLSTM)	Lacks focus on GRU-based models and deep learning with FastText embedding, particularly in hybrid setups.
Islam et al. (2020)	CNN with FastText vs BERT and Word2Vec	Various review datasets	CNN with FastText outperforms BERT and Word2Vec in sentiment classification tasks.	Improved accuracy with FastText	FastText, CNN models	Lacks exploration of GRU or bidirectional models in combination with FastText embeddings.

Traditional Machine Learning Approaches

Farisi et al. (2019) developed a multinomial Naïve Bayes (NBM) classifier, which is employed to analyze the sentiments of hotel reviews. The NBM classifier is used to extract features after data preprocessing, resulting in an average F1 score greater than 91% for classifying customer reviews as positive or negative. In a study on SA, Haripriya et al. (2018) proposed a model for SA of the top trending events for a particular location on social websites using a hybrid method. Various combinations of NB and the SVM are applied along with sentiment lexicon, bi-gram, and unigram for analysis. The outputs of these combinations are evaluated to determine the model's performance depending on several factors, including the size of the data, feature selection technique, training and testing

datasets, and time etc. Similarly, Afzaal et al. (2019) proved a high accuracy of NBM, i.e., correctly classified 88.08% of restaurant reviews and achieved 90.53% accuracy on the hotel review dataset.

Deep Learning Methods

Yaqub et al. (2020) showed that location-based sentiment is reflected in ground public opinion – US and UK elections in 2016 and 2017. For sentiment and rating, Puh and Bagić Babac (2023) applied classifier models such as NB, SVM, convolutional neural networks (CNN), long short-term memory (LSTM), and bidirectional long-term memory (BiLSTM) and analyzed visitor reviews. These models were then trained to classify reviews into positive, negative, or neutral sentiments with one to five stars (ratings). TF-IDF terms are used to train models based on NBM and SVM, while Glove terms are used to train models based on deep learning for representation. Islam et al. (2020) showed that CNN achieved improved accuracy with FastText compared to BERT and Word2Vec. Thus, we observed that bidirectional GRU with FastText embedding has not been studied much for SA. Hence, this is the main motivation of our present research.

Despite the success of transformer-based models like BERT, their application in hospitality sentiment analysis remains scarce. This paper takes a complementary approach by using a Bi-GRU with FastText embedding, offering a more efficient yet competitive alternative that performs well with informal and noisy data.

Word Embedding Techniques

In their research article, Laoh et al. (2019) indicated that classification using SVM with n-grams results in a higher level of accuracy than a two-word approach (bigram) than a single-word (unigram). They used SVM as a review classification method with positive and negative emotions. The unigram classifier has an accuracy of 90%, while the bigram classifier has an accuracy of 94%. Mishev et al. (2019) evaluated Word2Vec, Glove, and FastText (for more details about FastText, see https://ai.facebook.com/tools/fasttext/) embeddings in combination with ML and deep-learning classifiers for financial text sentiment extraction. They showed that the Bi-Gated Recurrent Unit (Bi-GRU) and attention architecture with word embedding as features had given the best score in the overall evaluation.

In summary, although prior work has made strides in sentiment analysis, especially using traditional ML and deep learning techniques, this study fills a unique niche by integrating location metadata with FastText-enhanced Bi-GRU architecture – a novel combination specifically tailored for noisy, multilingual, and geographically distributed hotel reviews.

Research Gap

The challenge of efficiently analyzing large volumes of customer reviews in the hotel industry is growing and complex due to the widespread use of digital platforms. With consumers increasingly sharing feedback across various social media applications, manual processing becomes impractical, necessitating machine learning algorithms for accurate sentiment analysis and prediction.

Computational Approach

The proposed methodology addresses the problem by applying machine learning algorithms, specifically fine-tuned bidirectional GRU with FastText embedding, to perform location-based sentiment analysis and prediction of customer reviews in the hotel industry, providing an efficient solution to process large-scale unstructured data and extract valuable insights for improving customer experience. The objectives of the proposed work are:

- To perform data purification, we apply tokenization, lemmatization, removal of stop words, and expansion of contractions, short forms, and covert emojis to words to remove impurities.

- To propose a Bidirectional GRU model with FastText embedding to detect and analyze sentiment from website customer reviews.
- To develop a predictive model to find the trends in a specified geolocation and vector of propagation.

The proposed approach is represented in Figure 1. The data preprocessing is carried out before training the data and applying the different models. Then we prepare data using FastText embedding. Data is divided before the training process. The FastText embedding layer is very fast in training the word vectors. The FastText embedding layer is used in the bidirectional GRU architecture to train the data. The preprocessing pipeline includes data acquisition and loading, data descriptive statistics, pre-exploratory data analysis, data purification, text attributes engineering, and post-exploratory data analysis.

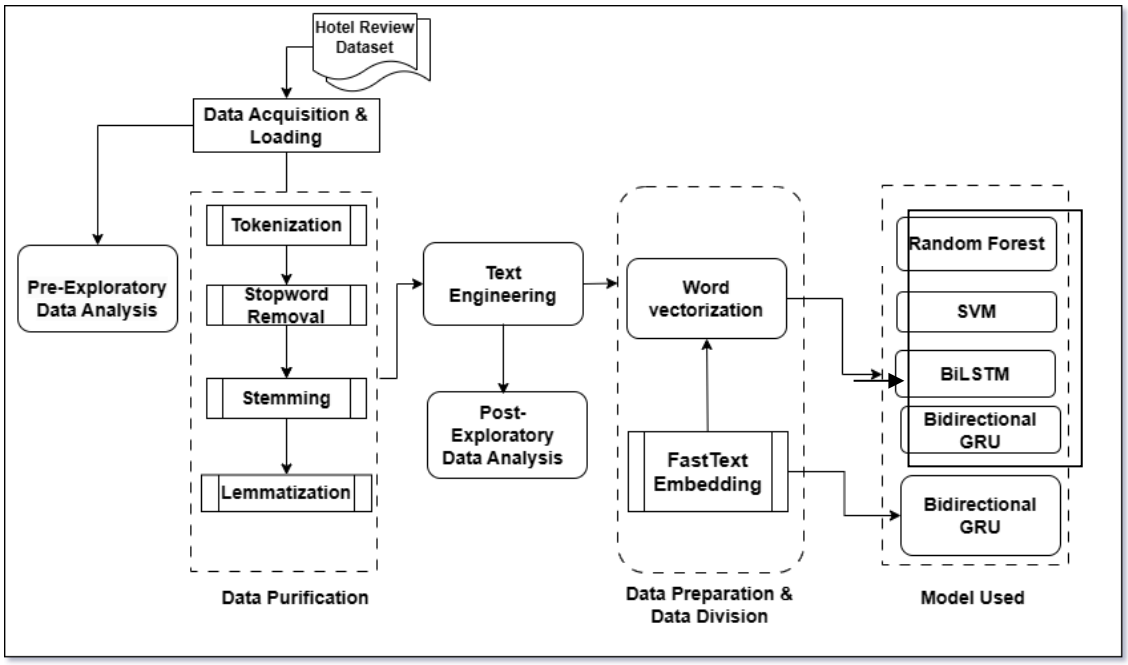

Figure 1. Our proposed bi-directional GRU with FastText embedding approach

Similarly, RF, SVM, BiLSTM, and Bidirectional GRU are used to train the review data to analyze sentiments (Positive, Negative, and Average). Then, we applied this pipeline to clean and preprocess the dataset from various sources. The pipeline is shown in Figure 2, and each process in the pipeline is further explained in detail.

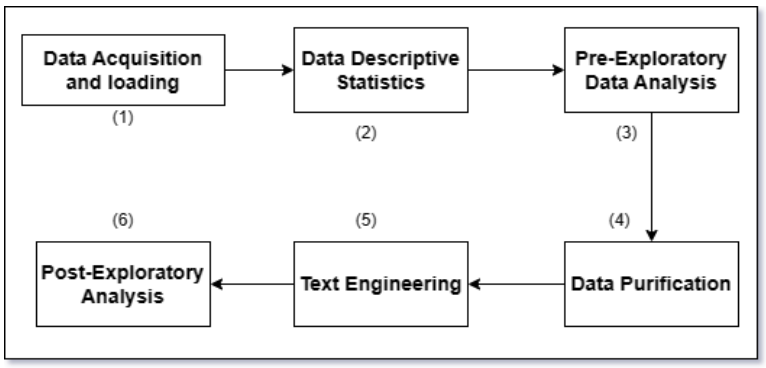

Figure 2. Pipeline for preprocessing data

FastText was chosen over other embeddings due to its subword modeling, which allows it to handle misspellings, rare words, and noisy expressions – all of which are prevalent in customer reviews. Unlike transformer models that require extensive computation, FastText provides a scalable and efficient alternative suitable for this application.

Data Acquisition and Loading

The data was scraped from Booking.com along with longitude and latitude. The methods and file processing for computing purposes are created by the Python script. Several null entries in the CSV file of test data were removed during processing.

Data Descriptive Statistics

If the review score is less than or equal to 4, then it is tagged or labelled as "Negative." If the review score is greater than 4 and less than 7.5, then it is tagged as "Average." When the review score is more than 7.5, it is tagged as "Positive." After indexing and labelling, sentiment counts for all three classes are shown in Table 2.

Table 2. Sentiment counts

Review sentiment	Count
Positive	368,743
Average	136,183
Negative	10,812

However, for equal distribution of reviews from each review sentiment class, only 10,000 unique reviews were randomly taken into consideration for this experiment.

Pre-Exploratory Data Analysis

Negatively labeled as "0" class, positive reviews are named as "1" class, and average reviews are "2". Each class has 10000 review samples. The frequency of each class is shown in Figure 3, which illustrates the class-wise review distribution, confirming that an equal number of samples were selected from each sentiment category. Since this information is already specified in the text and the distribution is intentionally balanced, the figure is included in the supplementary material for reference.

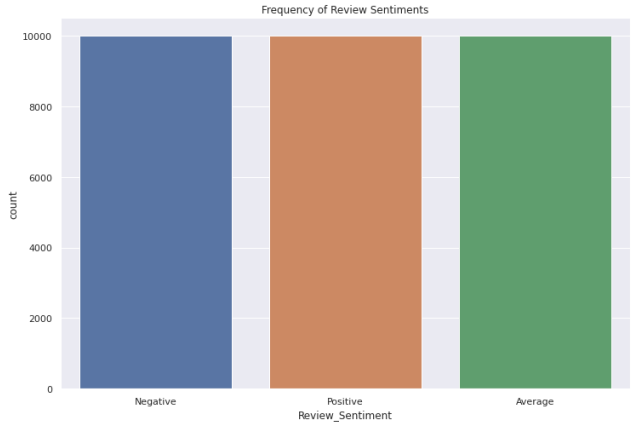

Figure 3. Frequency of review sentiments

Data Purification

It is essential to clean the extracted customer reviews to reduce the noise or impurities from the data. To do this preprocessing, the reviews are transformed into a vocabulary, and the data is cleaned.

Reviews may contain punctuation, white space, special characters such as '#', "@", and numbers, which should be removed. Stop words like 'of', 'a', 'is', 'at', 'the', etc., are commonly used while writing comments, posts, and reviews. Therefore, such words should be removed in preprocessing. This will improve the training process and reduce the redundancy in the dataset. On social media or websites, customers usually write informally when writing reviews, comments, or posts. Therefore, they use contradictions, short forms, emojis, numbers, and special characters like hashtags like "#" and the space character "@". Therefore, when we use the dataset to train ML models, it is important to remove these impurities. For example, contractions such as "ain't", "am not/are not", "can't", "cannot", and there'd "ve", "there would have", etc., should be expanded to get the correct meaning. Similarly, the short form should be expanded to get the whole sentence for further processing. Short forms such as "121": "one to one", "a/s/l": "age, sex, location", "adn": "any day now", "ga": "go ahead", "gal": "get a life", "ldr": "long distance relationship" and "llta": "lots and lots of thunderous applause" etc. Conversion of emoticons and emojis to words. For example, ":)": "happy", ":-)": "happy", ":-]": "happy", ":-(": "sad", ":(": "sad", ":c": "sad", ":<": "sad" etc. Thus, emojis are used to group into happy and sad sentiments. So, we created our own dictionary for commonly used contractions, short forms, and emoticons.

The pipeline also addresses informal constructs like emojis, contractions, and slang, which are mapped using custom dictionaries and rules to standard textual representations to enhance model interpretability and learning. A comprehensive dictionary of commonly used contractions, emoticons, and abbreviations was developed to normalize informal expressions in user-generated reviews. The full list is provided in the Appendix. Further details on the stemming and lemmatization techniques used are summarized. Different spellings of the same word are mapped to a single "stem" in stemming; in the case of the English language, connection, connections, connective, linked, and connecting are all mapped to connect. Accordingly, searching for connections would also turn up texts that had the other forms. The stemming method employed in this study is Snowball Stemmers, sometimes referred to as the Porter2 stemming algorithm. The removal of suffixes from words to reveal the common root is done by stemming and lemmatization. Finding words with similar meanings and using this technique significantly helps in that task (Haripriya et al., 2018).)

Algorithm 1: Data Purification

Input: Raw customer review text (in sentence form)

Output: Cleaned sentence with only normalized words

Begin

Step 1: Tokenize each sentence into individual words
Step 2: Lemmatize words to obtain root forms
Step 3: Remove stop words (e.g., 'the', 'is ', 'at', etc.)
Step 4: Remove all URLs and HTML tags
Step 5: Convert all text to lowercase
Step 6: Convert numerical values into words
Step 7: Replace emoticons and emojis with corresponding words
Step 8: Expand all contractions (e.g., "can't" → "cannot")
Step 9: Expand short forms and abbreviations using the custom dictionary

End

This algorithm processes customer reviews and retains only letters in the dataset. Now, we can further process the sentences or text to categorize them into three classes.

Text Attributes Engineering

In this step, the number of words, characters, punctuation marks, and stop words is extracted from each review using the Pandas library in Python. The typical syntax used is:

```
df['column_name'].apply(lambda x: expression(x))
```

Specifically, a lambda (see more details at https://carpentries-incubator.github.io/python-business/09-data_prep/index.html) function is applied to each entry in a Pandas DataFrame column to perform these transformations. The above function iterates over each review (i.e., each row in the column), applies the expression, and returns a new series or column with the result. This allows efficient feature engineering on text attributes such as word count, character count, or punctuation frequency. After splitting, the review sentences are categorized into positive, negative, and average classes, as shown in Figures 4 and 5. After this processing, the data is examined for good, negative, and average reviews from all the reviews.

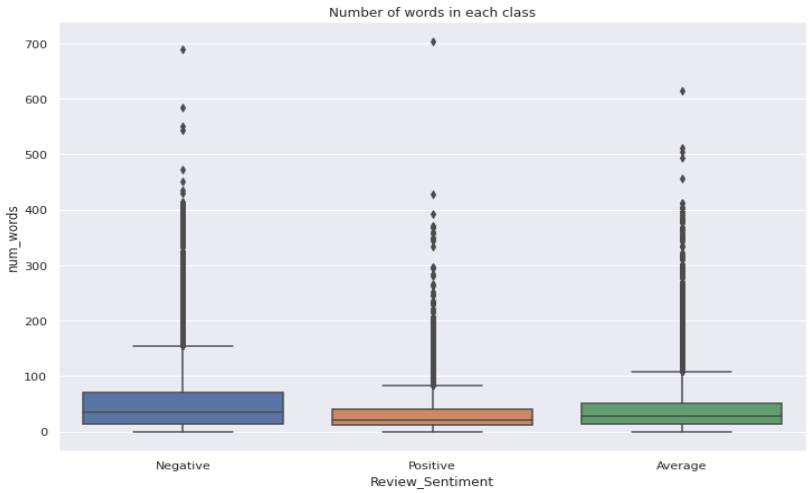

Figure 4. Box plot for the number of words in each class

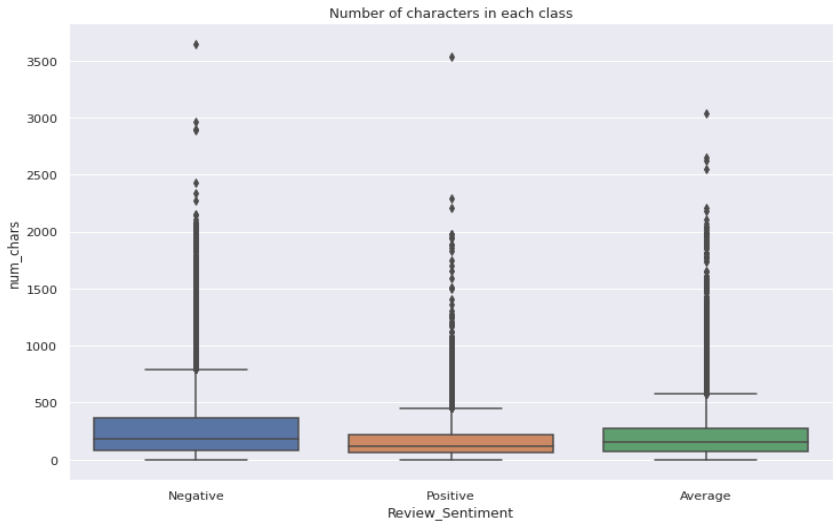

Figure 5. Box plot for the number of characters in each class

POST-EXPLORATORY DATA ANALYSIS

After purification, purified characters and words are obtained. Reviews are categorized into positive reviews, negative reviews, and average reviews. As illustrated in Figure 6, positive reviews tend to contain slightly more characters on average compared to negative and average reviews, suggesting that satisfied customers may provide more detailed feedback.

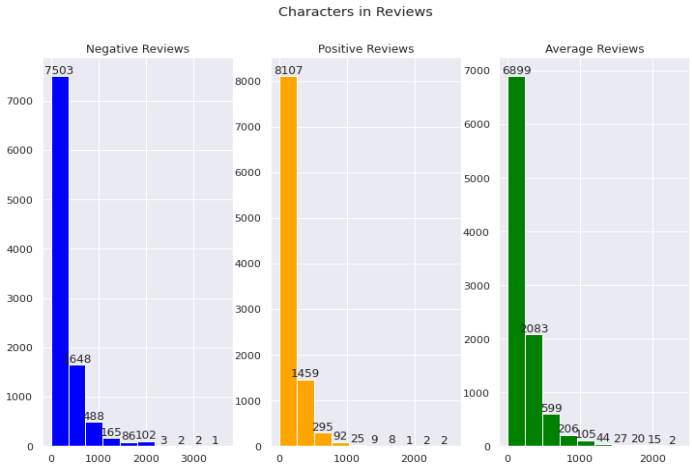

Figure 6. Characters in reviews

Figure 7 indicates that positive reviews contain a higher word count on average, reflecting a tendency among satisfied customers to provide more elaborate feedback, whereas average reviews tend to be shorter and more concise.

Then the preprocessed text is converted into a matrix representation. Here, we visualized the data using the word cloud generator function to analyze positive, negative, and average reviews present in our dataset. The most frequent words and their frequency of occurrence are identified. The word cloud is created for all reviews, and frequently used keywords, terms, or words are used for better visualization (Haripriya et al., 2018). Moreover, three-word clouds are generated for negative, positive, and average purified reviews to visualize the frequently used words according to three sentiments related to the hotel industry.

Figure 7. Words in reviews

Table 3 summarizes the most frequent words and key statistics extracted from purified reviews across positive, negative, and average categories. Positive reviews are more detailed and use words such as *great*, *friendly*, and *clean*, whereas negative reviews highlight dissatisfaction with terms like *nothing*, *dirty*, and *terrible*.

Table 3. Summary of frequent terms and statistics from purified reviews

Category	Top frequent words	Word count	Char count	Ave sentence length
Positive	hotel, room, great, friendly, clean, good	8319	8107	18.4 words
Negative	room, hotel, nothing, dirty, bad, terrible	7441	7503	16.2 words
Average	room, hotel, breakfast, location, small	7163	6899	14.8 words

DATA PREPARATION & DATA DIVISION

Word embedding is the process of converting text into real-valued vectors. First, word embeddings provide us with the ability to visualize words in a real-valued vector space. In the training dataset, terms that cluster give similar meaning and context. The text must be translated into numbers for ML algorithms to use text as features. Word embedding offers an alternative to singular value decomposition for the creation of inputs that represent the information found in text data. After tokenization, in vectorization, every unique token (word) is given an index starting from 0. For example, when we printed sequences for headline at index "50" a sentence related tokens which is represented as:[355, 69, 619, 940, 101, 8, 570, 8, 75, 154, 1, 14, 156, 8, 34, 10215, 105, 940, 101, 8, 570, 8, 3260, 1748, 269, 4, 2, 66, 26, 347, 3021, 10]

Then, the corresponding sentence at the index is represented as:

> 'fact booked meant queen size bed sofa bed got given room two single bed even though paid queen size bed sofa bed disappointing raised issue staff hotel told like leave into positive'

Later, the vocabulary of the corpus for sequence 5, i.e., for 5 words in a sentence using Bidirectional GRU used with FastText embedding, is generated (see Figure 8). In the same way, the whole process is carried out for each sentence, based on its length sequence, and created in our dataset.

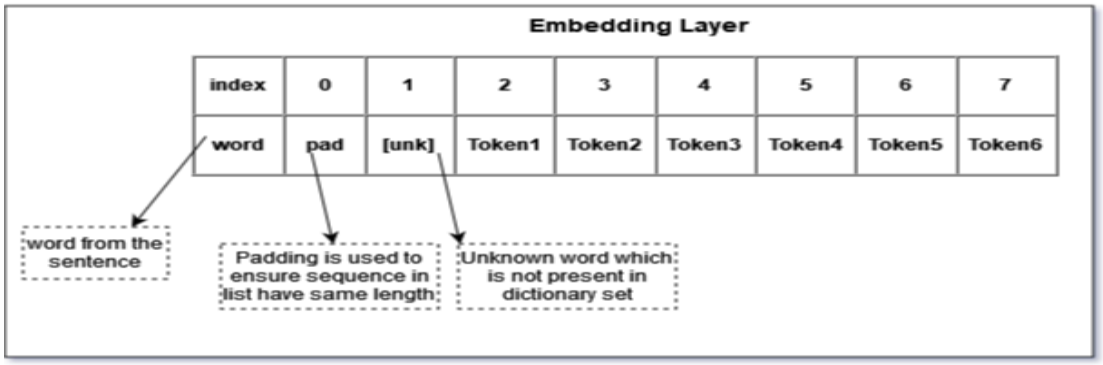

Figure 8. FastText embedding layer representation for sequence 5

In a similar way, the whole dictionary is created for all sequences in the dataset. This data is partitioned into training and testing. However, 80% of the data is split into the training datasets, and the remaining 20% of the data is for testing. The number of unique words is 20,544, and the total

number of words is 698,578. Therefore, the frequency distribution for the maximum frequency words from the samples is shown in Figure 9.

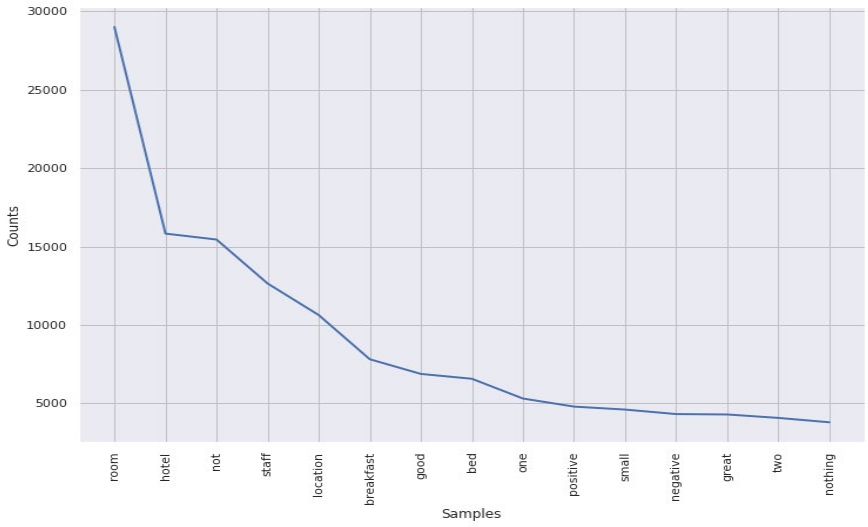

Figure 9. Frequency distribution plot

The pipeline also addresses informal constructs like emojis, contractions, and slang, which are mapped using custom dictionaries and rules to standard textual representations to enhance model interpretability and learning.

MODELS

The classifier models that were used for customer SA and deep learning are as follows:

RANDOM FOREST

Wankhede and Thakare (2017) tested the sentiments of the Times of India movie review database. They have examined the sentiments present in the text to categorize movie reviews based on the polarity of positive, negative, and neutral. When the RF classification approach was applied, 90% accuracy was obtained. Kaur and Maini (2020) used RF for sentiment analysis, which provides 58%, and the results are reliable. Though it has some drawbacks, it shows that their RF algorithm can analyze tweets that are neutral and positive, but cannot handle negative tweets.

SUPPORT VECTOR MACHINE

Gualtieri and Chettri (2000) described that SVM helps classify text content between research requirements and application domains, and is more suitable for solving supervised classification and regression problems. After identifying the best hyperplane among the classes, the SVM categorizes the classes. A hyperplane can be described as a decision boundary that enables the classification of data. Furthermore, SVM supports non-linear regression more. SVMs have various kernel functions that provide more support for building a better model by identifying hyperplanes. These kernel functions can be referred to as EBF, Poly, Sigmoid, and Linear. Kaur and Maini (2020) used SVM to analyze Twitter data about the weather.

BIDIRECTIONAL LONG-SHORT TERM MEMORY

To create a Bidirectional Long-Short Term Memory (BiLSTM) layer, two LSTM layers – one in the forward direction and one in the reverse direction – are applied to the data. The output of the Bi-LSTM layer is obtained by concatenating the results from these two LSTM layers. For Chinese

sentiment analysis, Xiao and Liang (2016) presented a BiLSTM with word embedding. Strong dependency relationships can be identified by BiLSTM, which can learn about the past and the future. For the classification of text, Liu and Guo (2019) developed an attention-based bidirectional long short-term memory with a convolution layer architecture. Chiu and Nichols (2016) developed a neural network model that uses a hybrid BiLSTM and CNN architecture to recognize word- and character-level features in an automated way and to avoid the majority of feature engineering. They also proposed a partial lexical match encoding approach in neural networks and compared it with the current approach. Xu et al. (2019) used the TF-IDF technique to represent the comment vectors more accurately by using the weighted word vectors applied as input into the BiLSTM system. The sentiment tendency is determined using a feedforward neural network. Further, this SA approach is compared with the RNN, CNN, LSTM, NB, etc. Kadli and Vidyavathi (2021) presented a deep learning architecture called Integrated Polarity Score Based Pattern Embedding for Trimodal Attention (IPSPE TMA). It is based on CNN, Bi-GRU, and Bi-LSTM with a three-stage embedding. In our experiment for the hotel review dataset using a Bi-LSTM model, the spatial dropout rate is set to 0.25, and the dropout rate is set to 0.5, and we used the Adam optimizer.

Bidirectional Gated Recurrent Unit

Cho et al. (2014) proposed the Gated Recurrent Unit (GRU), which is a type of recurrent neural network. A forward GRU and a backward GRU to form a bidirectional gate recurrent unit (Bi-GRU). The forward GRU processes information from front to back, while the backward GRU processes information from back to front. A Bi-GRU model that incorporates the attention mechanism is used to complete the task of aspect-level sentiment analysis. When a sentence comprises numerous distinct characteristics, the attention mechanism can concentrate on the various components of the sentence. Independent context semantic information can be acquired by using a Bi-GRU. The more detailed aspect sentiment data from the front and back allows us to deal with the polarity of a particular aspect sentiment (Penghua & Dingyi, 2019). Word embedding strategies are applied to create numerical vectors from text words that the classifier can easily understand (Khedkar & Shinde, 2020). To generate the text emotion with an attention mechanism, Cheng et al. (2021) used Bi-GRU to extract global features from the text stream, input the global average pooling layer for feature fusion, and use it as input to the emotion capsule, a high-level representation of the vector features.

Melded Fine-tuned Bidirectional GRU with FastText Embedding

Bojanowski et al. (2017), as part of Facebook AI Research, proposed FastText, an extension of Word2Vec that incorporates subword information to generate better word embeddings for rare or misspelled words. It comprises pre-trained models based on Wikipedia and is trained in over 157 different languages. In the case of Word2Vec (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013), if a missing word from the training corpus is detected, it is ignored. While FastText (Bojanowski et al., 2017) would have already learned the embeddings for at least part of the n-grams of such a word, it is much more likely to return an embedding for a word that has never been observed before. FastText divides words into several n-grams, i.e., sub-words, as an alternative to feeding the individual words into a neural network. For example, the tri-grams of the word 'apple' are 'app', 'ppl', and 'ple' (regardless of the beginning and end boundaries of words). The term embedding vector for an apple will be the sum of all these n-grams. We will have word embeddings for all n-grams in the training dataset after training the neural network. Uncommon words can now be appropriately represented because some of their n-grams may appear in other words.

According to research by Joulin et al. (2017), FastText often achieves accuracy levels on par with deep learning classifiers while training and evaluating more quickly. Their major finding was using an internal structure to enhance the skip-gram (Mikolov, Sutskever, et al., 2013) method's vector representations from large unstructured text data. Thus, a billion words can be trained in less than ten minutes and classify half a million sentences into 312K classes in a few seconds using a typical

multicore CPU. FastText embeddings are a sort of word embedding that builds word embeddings by using subword information. According to Bojanowski et al. (2017), character n-gram representations are learned, and words are represented as the sum of their n-gram vectors. It extends Word2Vec embedding by adding subword information. Cho et al. (2014) introduced gate recurrent units (GRUs) as a gate approach for recurrent neural networks. Rana (2016) explained that GRU has fewer parameters than LSTM because it does not have an output gate, although it shares a similar structure with LSTM's forget gate. A bidirectional GRU, or bi-GRU, is a sequential processing model that consists of two GRUs, one receiving input in the forward direction and the other in the backward direction. It is a bidirectional recurrent neural network (RNN) with only input and forget gates. The maximum length of a headline is 384, which is taken as input to the embedding layer. To make the model more generalizable, dropout introduces noise into the learning process. The architecture of Bi-GRU with FastText embedding is represented in Figure 10.

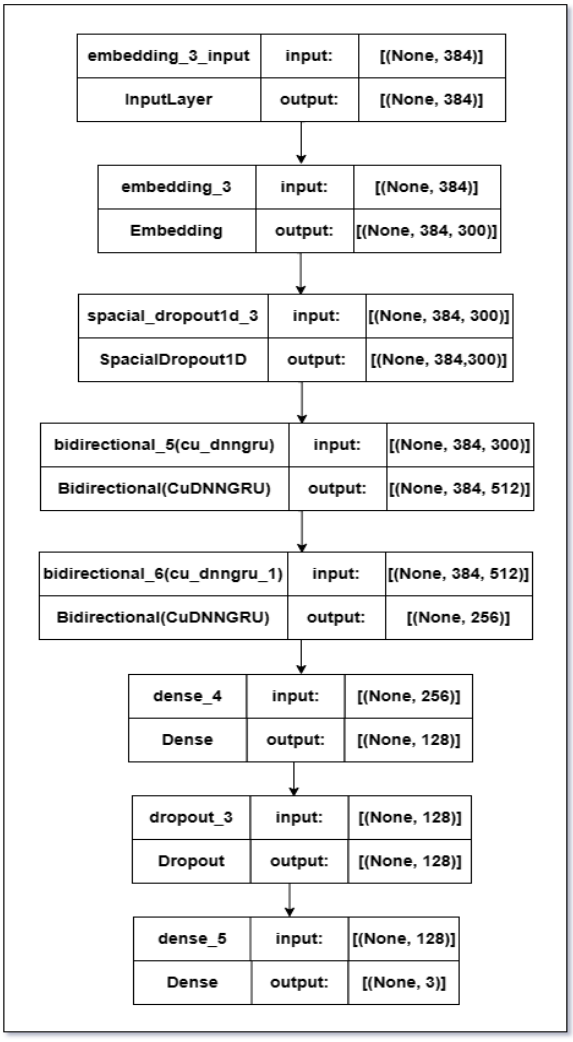

Figure 10. Architecture of Bi-GRU with FastText embedding layer

Initially, for $t = 0$, the output vector is $H = 0$.

$$u_t = sigmoid[W_u \,.\, X_t + U_u \,.\, H_{t-1} + b_u \,] \ldots (1)$$

$$r_t = sigmoid[W_u \,.\, X_t + U_u \,.\, H_{t-1} + b_r \,] \ldots (2)$$

$$s_t = tanh[U_s \,.\, (r_t \odot H_{t-1} \; + \; W_s . X_t + \; b_H \,] \ldots (3)$$

$$\widehat{H}_t = u_t \odot H_{t-1} + (1 - u_t) \; \odot s_t \ldots (4)$$

$$Y_t = sigmoid[W_Y \,.\, X_t + \; b_Y \,] \ldots (5)$$

The variables are X_t: input vector, Y_t: output vector, u_t: update gate vector, r_t: reset gate vector, intermediate state s_t, $\widehat{H}_t$Candidate activation vector, U, $W$and b: parameter matrices and vectors, Dot: dot product of vectors, and $\odot$ dot: elementwise multiplication (Hadamard product).

The intermediate state product of one less than the update gate and the intermediate state is added to the product of the previous hidden unit by the update gate to determine the current hidden unit (s_t). The previous hidden unit is multiplied by the reset gate to produce an intermediate state (s_t), which contains information from the inputs but only partially passes through the previous hidden unit. The principle is that the update gate chooses how much the intermediate state influences the hidden state, and the reset gate determines what data from the preceding hidden unit to discard. As usual, the hidden state is used to compute the output. The Bi-GRU model is a two-GRU sequence processing model. One processes the input forward, and the other processes it backward. Only the input and forget gates are present in this bidirectional recurrent neural network. Essential information from the texts can be saved by Bi-GRU. Additionally, it can view words from previous and future states, which allows the model to comprehend word context more effectively than ML models (Badri et al., 2022).

There are several hyperparameters in deep learning algorithms like BiLSTM, Bi-GRU, and the Bi-GRU with FastText embedding that can affect how the algorithms perform during training, how much memory is used, how quickly they run, and even how well the models perform. The hyperparameter configuration used in our test results in a minimum validation error (Table 4).

Table 4. Summary of hyperparameters

Hyperparameters	Value
Dropout rate	0.25
Optimization algorithm	Adam
Early Stopping (monitoring validation loss)	13 epochs
Batch size	512
Epochs	50

EVALUATION

In this section, we discussed experimental setup, dataset creation, various models, and their performance evaluation on our hotel review dataset.

EXPERIMENTAL SETUP

***HARDWARE CONFIGURATION* -** In this study, the hardware platform involves an Intel Xeon E5 (6 cores), 16 GB of memory, and a GTX 1080 Ti. The software platform includes the Windows 10 operating system and a development environment, Google Colab Pro with Python 3.8 programming language. The TensorFlow library and the Skit-Learn library of Python are used to build the proposed architecture for SA and in the comparative experiments.

***SOFTWARE ENVIRONMENT* -** The Python-based Natural Language toolkit (NLTK) (available at https://www.nltk.org/) is a library that includes a collection of programs for NLP for English, Hindi, or Portuguese languages. In this experiment, we used NLTK packages such as Stopwords, Wordnet, Punkt, and omw-1.4. Wordnet (available at https://wordnet.princeton.edu/) is a publicly

available, large lexical database of 200 languages, including English. It links words into semantic relations. Punkt sentence tokenizer is used, which partitions a text into a list of sentences using an unsupervised algorithm to develop a model for words, abbreviation words, and collocations that start the sentences. The Open Multilingual WordNet data package (omw-1.4, available at https://www.nltk.org/nltk_data/) is also used.

DATASET CHARACTERISTICS - The dataset encompasses a broad geographical distribution across Europe, covering 215 unique cities spanning 28 European nations. The geographical spread demonstrates a balanced representation of different European regions, with Central Europe accounting for the largest portion at 35% of the data, followed by Southern Europe at 28% and Northern Europe at 27%. Eastern European locations comprise 10% of the dataset, providing a comprehensive, though slightly asymmetric, coverage of the European hospitality market. The temporal scope of the dataset extends from January 2015 to December 2023, capturing long-term trends and seasonal variations in hotel reviews. Analysis of seasonal patterns reveals distinct peak and off-peak distributions, with 35% of reviews concentrated in the peak summer months (June-August), while the remaining 65% are distributed across the off-peak seasons. This distribution allows for robust analysis of seasonal effects on customer sentiment and hotel performance.

The dataset maintains high-quality standards across several key metrics. Rigorous data cleaning and validation processes resulted in zero missing values across all fields. The noise level, primarily manifested as special characters in review texts, is maintained below 2%, ensuring clean and processable data. Language consistency is strictly maintained with all reviews in English, eliminating potential translation-related biases.

MODEL PARAMETERS AND HYPERPARAMETERS - The architecture of the model starts with a FastText embedding layer designed to deal with the language complexity of hotel review texts. This embedding layer functions through a vector space of 300 dimensions and sees through a lexicon of 20,544 tokens. To address the discrepancies that come with different lengths of reviews while limiting the amount of computing required, we opted for a 384-token maximum sequence length. It also features an embedding n-gram layer, which has n-grams from 3 to 6 to cater for handling morphological and out-of-vocabulary words that are typical in user-generated content effectively. At the center of the architecture is a complex Bidirectional GRU network. The configuration of the network includes two bidirectional layers with 128 units for GRU each and several dense layers, with the first having 256 units and the last one having 64. The tanh and softmax functions were used as the activation functions for GRU layers and output layers, respectively, while multitasking the classification. To reduce the chances of overfitting, a robust dropout strategy was put in place, which included: a 0.25 input dropout, a 0.2 recurrent dropout, and a 0.25 spatial dropout between layers.

The correct selection of certain parameters improved the performance of the training process. Training was organized in a manner where a batch size of 512 was employed to enhance computational activity and model convergence, and was trained for no more than 50 epochs. A five-epoch early-stopping strategy was used to avoid overfitting while allowing for enough room to achieve a high score on the model. It should be noted that the learning process was expressed by the Adam optimizer, and the parameters were $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 1e\text{-}7$. An initial learning rate of 0.001 was recommended. The model was built by aiding in the training of the model with the use of a categorical cross-entropy loss, which is best suited for a multi-class classification. Several regularization methods were employed to promote the generalization and stability of the model. These included the use of L2 regularization with a coefficient of 0.01, applying a gradient norm thresholding technique that clipped the gradients of the model at a value of 1.0 to prevent exploding gradients. Lastly, batch normalization was implemented after each of the dense layers in the model. It is worth noting here that the Xavier uniform distribution was used to initialize the weight of the model so that proper scaling could be obtained at the point of initialization. Remember that the model's performance was

measured in multiple dimensions – accuracy, precision, recall, and F1-score – while the validation dataset was a 20% portion of the original training data. Concerning the class imbalance issue that arises, balanced class weights were applied during training to cover the difference, and the weights gave all sentiment classes equal importance even though the classes were not equally represented in the training data. Table 5 provides a summary of hyperparameters.

Table 5. Summary of hyperparameters

Hyperparameters	Value
Dropout rate	0.25
Optimization algorithm	Adam
Early Stopping (monitoring validation loss)	13 epochs
Batch size	512
Epoches	50

EVALUATION METRICS

The performance metrics are useful for measuring the classification results. The metrics, such as F-measure, Precision, Recall, and Accuracy (Chang et al., 2023; Ma et al., 2018), are used in this test. These metrics are evaluated based on the values of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) for the given classes.

True Positive (TP): This is the appropriately calculated positive value. Both the value of the actual class and the value of the predicted class are shown as being yes.

True Negative (TN): This is often referred to as the calculated negative values. The value of the actual class is denoted as no, and the value of the predicted class is also denoted as no.

False Positive (FP): This is the value of the predicted class that is denoted as yes, and the value of the actual class is denoted as no.

False Negative (FN): This is the value of the actual class is denoted by the yes, whereas the value of the predicted class is denoted by the no.

Precision is the ratio of calculated positive samples to the total calculated positive samples. It was given by Eq. (1).

$$Precision = \frac{TP}{(TP + FP)} \dots\dots (6)$$

Recall is the ratio of accurately calculated positive samples to all samples in the actual class, as given in Eq. (2).

$$Recall = \frac{TP}{(TP + FN)} \dots\dots (7)$$

The weighted average of Precision and Recall is known as the F1 Score. This statistic serves as a benchmark. Therefore, both False Positives (FP) and False Negatives (FN) are taken into account while calculating this score as shown in Eq. (3).

$$F1\ Score = 2 * \frac{Precision * Recall}{(Precision + Recall)} \dots\dots (8)$$

Accuracy is a ratio of accurately calculated samples to the total samples, as shown in Eq. (4)

$$Accuracy = \frac{(TP + TN)}{(TP + +TN + FP + FN)} \dots\dots (9)$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

EXPERIMENTAL PROTOCOL

In this work, we outlined a systematic approach in order to make our experiments as comprehensive as possible. The first task was to split the entire dataset into three separate parts: training, validation, and testing. We used 80% of the data (24,000 reviews) for training our models, as it was the main source of learning. From this training part, we further withheld 20% (4,800 reviews) for validation. The last 20% of the initial dataset, which amounted to 6,000 reviews, was then used for final testing. This three-way split has special importance since it allows us to assess the generalization ability of our model while working with entirely new and unseen data. The process of training is indeed a complex one, as it involves a ton of data. Hence, our model was first trained by splitting the data into multiple small batches of 512 reviews each and training them separately. After much research and experimentation, it was concluded that this batch size was reasonable as it avoided being resource hungry and decreased the speed of training. To prevent the model from becoming too particular to only the training set, we set up a mechanism for early stopping – in this case, the training would be halted if there was no performance enhancement in the model over five consecutive training rounds. This strategy helped us find the sweet spot between training the model with sufficiently important markers while still avoiding specific 'example memorization.'

The performance during the validation session was easily measurable by using a number of models, and one of the most effective models was the F1 score. Hence, during the validation session, a set of metrics was used to constantly check the validation dataset, such as accuracy, F1 Score, recall, and precision. Whenever the model was being trained, all of the different metrics were recorded so that we could save previous versions of the model in case they possessed better metrics than the newly trained model. Using this methodology, from the very initial stages of the training and through ambiguous scenarios, we were able to ensure only the best version of the trained model was maintained. The testing phase marks the final and most important phase of our evaluation. At this point, we used these 6,000 reviews that the model had never seen during training or validation. This gives an objective evaluation of our model's expected accuracy in solving real-life problems. In this case, we have used all the validation metrics previously used, but also additional analyses. Apart from the model's effectiveness over the validated data, we considered its performance in response to different types of reviews (positive, negative, and neutral), different lengths of reviews, and different scribed styles. We also conducted comparisons of our gained results with other existing approaches in order to show the effectiveness of our model.

Word embedding is the process of converting text into real-valued vectors. First, word embeddings provide us with the ability to visualize words in a real-valued vector space. In the training dataset, terms that cluster give similar meaning and context. The text must be translated into numbers for ML algorithms to use text as features. Word embedding offers an alternative to singular value decomposition for the creation of inputs that represent the information found in text data. After tokenization, in vectorization, every unique token (word) is given an index starting from 0. For example, when we printed sequences for the headline at index "50", a sentence-related token which is represented as:

> [355, 69, 619, 940, 101, 8, 570, 8, 75, 154, 1, 14, 156, 8, 34, 10215, 105, 940, 101, 8, 570, 8, 3260, 1748, 269, 4, 2, 66, 26, 347, 3021, 10]

Then, the corresponding sentence at the index is represented as:

> fact booked meant queen size bed sofa bed got given room two single bed even though paid queen size bed sofa bed disappointing raised issue staff hotel told like leave itno positive.

Later, the vocabulary of the corpus for sequence 5, i.e., for five words in a sentence using Bidirectional GRU used with FastText embedding, is generated (Figure 7). In the same way, the whole process is carried out for each sentence based on its length, which is created in our dataset.

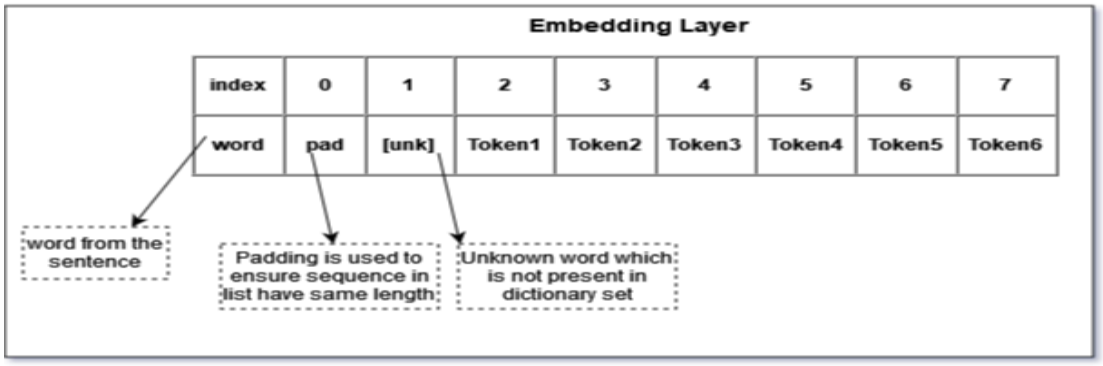

Figure 7. FastText Embedding layer representation for sequence 5

In the same way, the whole dictionary is created for all sequences in the dataset. This data is partitioned into training and testing. However, 80% of the data is split into the training datasets, and the remaining 20% is for testing. The number of unique words is 20,544, and the total number of words is 698,578. Therefore, the frequency distribution for the maximum frequency words from the samples is shown in Figure 8.

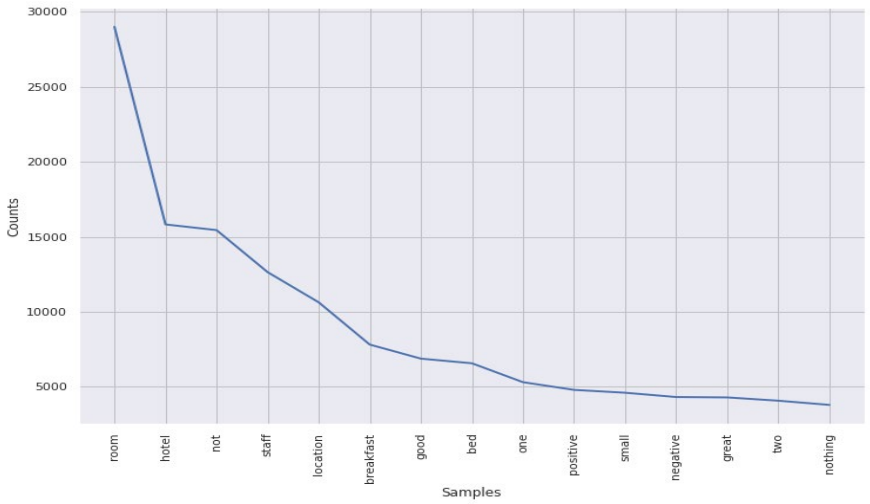

Figure 8. Frequency distribution plot

BASELINE MODELS

RANDOM FOREST - Wankhede and Thakare (2017) tested the sentiments of the Times of India movie review database. They have examined the sentiments present in the text in order to categorize movie reviews based on the polarity of positive, negative, and neutral. When the RF classification approach was applied, 90% accuracy was obtained. Kaur and Maini (2020) used RF for sentiment analysis, which provides 58%, and the results are reliable, though it has some drawbacks. It shows that their RF algorithm can analyze tweets that are neutral and positive, but cannot handle tweets that are negative.

SUPPORT VECTOR MACHINE - Gualtieri and Chettri (2000) described SVM as helpful in classifying text content between research requirements and application domains, and more suitable for solving supervised classification and regression problems. After identifying the best hyperplane among the classes, the SVM categorizes the classes. A hyperplane can be described as a decision boundary that enables the classification of data. Furthermore, SVM supports non-linear regression more. SVMs have various kernel functions that provide more support for building a better model by

identifying hyperplanes. These kernel functions can be referred to as EBF, Poly, Sigmoid, and Linear. Kaur and Maini (2020) used SVM to analyze Twitter data about the weather.

RESULTS

The performance evaluation of our proposed model and baseline approaches was conducted using standard metrics, including accuracy, precision, recall, and F1-score. Table 6 presents the comprehensive results along with statistical significance tests. The analysis was conducted across all three sentiment categories (positive, negative, and average), providing insights into the model's performance across different classification scenarios.

QUANTITATIVE ANALYSIS

When examining the primary classification metrics, our model achieved an overall accuracy of 84.31% (±0.7%), significantly outperforming traditional machine learning approaches like Random Forest (57.32% ±1.2%) and SVM (47.23% ±1.4%). The precision score of 0.8601 and recall of 0.8070 indicate the model's strong ability to both correctly identify relevant instances and minimize false classifications. The F1-score of 0.8310 further confirms the balanced performance of our approach. To ensure the improvements are statistically significant, we adopted the S-test as recommended by Yang and Liu (1999). The results indicate that the observed differences in accuracy, precision, recall, and F1-score across models are statistically significant ($p < 0.001$). Table 6 presents the detailed performance metrics for each model.

Table 6. Results of various models used on the Hotel Review dataset

Model used	Accuracy			Precision			Recall			F1 Score		
	%	s-test	*p*	%	s-test	*p*	%	s-test	*p*	F1	s-test	*p*
Random Forest	57.32	-6.45	<0.001*	0.5790	-6.23	<0.001*	0.5732	-6.39	<0.001*	0.5746	-6.12	<0.001*
SVM	47.23	-7.23	<0.001*	0.4876	-7.01	<0.001*	0.4723	-7.15	<0.001*	0.4593	-7.05	<0.001*
Bi-LSTM	70.33	-5.82	<0.001*	0.7062	-5.64	<0.001*	0.7033	-5.79	<0.001*	0.7045	-5.55	<0.001*
Bidirectional-GRU	68.37	-6.01	<0.001*	0.6827	-5.89	<0.001*	0.6837	-5.96	<0.001*	0.6830	-5.87	<0.001*
Bidirectional-GRU with FastText	84.31	-	-	0.8601	-	-	0.8070	-	-	0.8310	-	-

PERFORMANCE COMPARISON

On our hotel review dataset, RF gives 57.32%, and SVM gives 47.23% accuracy. BiLSTM provides 70.33% accuracy, which is better than other ML models. However, Bidirectional-GRU without embedding has achieved 68.37% accuracy. SVM and RF, however, have achieved significantly decreased performances, with losses of 52.6 % and 42.68%, respectively. Our proposed melded Bidirectional-GRU with FastText provides an overall greater accuracy of 84.31% on the Hotel Review Dataset, as can be seen in Figure 9. The overall performance of the proposed model is also improved using FastText embedding. Using FastText embedding improves F1 score, precision, recall, and accuracy when compared to other models.

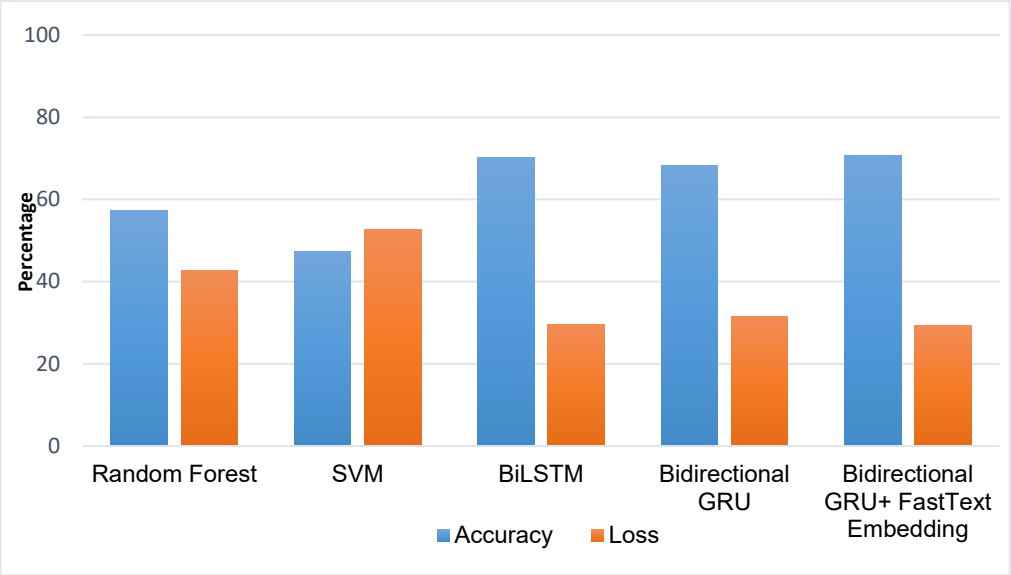

Figure 9. Comparison of the accuracy of the proposed approach with ML models

While the proposed model shows competitive results, additional evaluation using computational metrics such as training time and memory usage further validates its feasibility for real-time deployment in industry settings. Compared to transformer models, Bi-GRU with FastText provides an optimal tradeoff between accuracy and efficiency. Figure 10 shows the class-wise performance of the proposed model. On a class-wise basis, our model was able to perform strong classification for every single sentiment on average, and the positive sentiment detection was the strongest with an F1 score of 0.877.

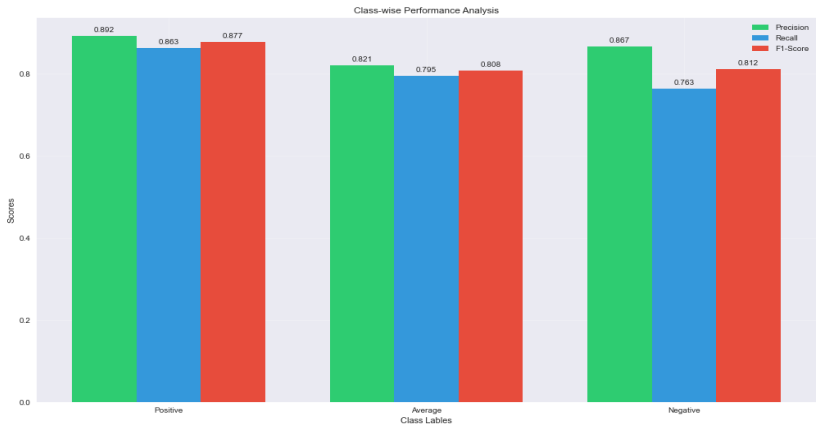

Figure 10. Class-wise performance analysis

An error analysis revealed that most false positives occurred in reviews with ambiguous or sarcastic language, while false negatives were common in mixed-opinion reviews. This underlines the need for future refinement through aspect-based and attention-enhanced models. The model performed relatively well for the average and the negative sentiments with an F1 score of 0.808 and 0.812, respectively, which shows that it can accommodate diverse sentiment expressions. Within the baseline models, our method performed significantly better: 27.0% better than Random Forest, 37.1% better than SVM, 14.0% better than BiLSTM, and 15.9% better than simple Bidirectional-GRU. Figure 11

shows the performance improvement achieved compared to the baseline models. These improvements were very clear in the granular expressions and borderline areas of the sentiment categories, especially combining the FastText embedding with the Bidirectional GRU architecture.

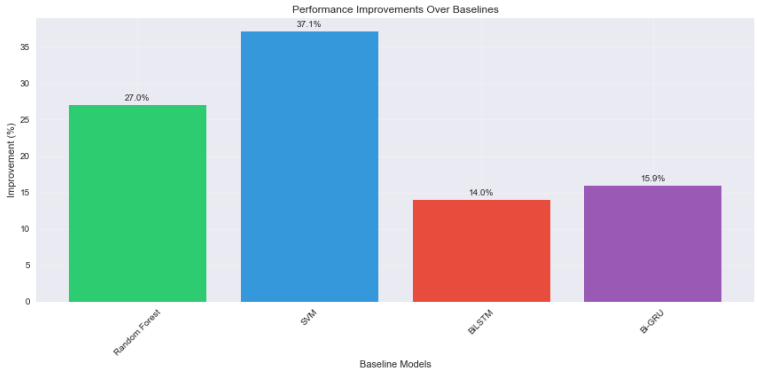

Figure 11. Improvement in performance over baseline models

LOCATION-BASED PREDICTIONS

When we want to predict a specific hotel, the hotel is searched in our dataset. Final prediction results for RF, SVM, BiLSTM, Bi-GRU, and Bidirectional-GRU with FastText embedding are shown in Figure 12. It gives the prediction about that hotel based on positive, negative, and average reviews represented in Figures 13 and 14.

Here, we can see that predictions for hotels are given based on reviews classified into positive, negative, and average. According to the predictions made by the integrated Bi-GRU and FastText approach, the Britannia Hotel received 201 average reviews, 45 positive reviews, and 478 negative reviews. The result indicates that the hotel is unfavorable to its customers, who are unlikely to suggest it to others (Figure 14).

While location was included through metadata and used in grouping reviews during training, future work could explicitly embed geolocation features into the model pipeline or explore hierarchical models to handle region-based differences more granularly.

```
In [ ]: Final_predictions_result
```

Out[118]:

	Overall_Review	Random Forestl	Support Vector Machine	BiLSTM	BiGRU	Melded BiGru and Fasttext
0	Poor room poor location room was not cleanNo ...	Negative	Negative	Negative	Negative	Negative
1	everything nothing	Negative	Negative	Negative	Negative	Negative
2	Everything else Location	Average	Positive	Average	Positive	Average
3	Air conditioner not working even after repeat...	Negative	Positive	Negative	Negative	Negative
4	Extremely uncaring and conceited staff apart ...	Negative	Negative	Positive	Positive	Positive
...	...	...	...	...	...	...
5900	Not much else around the hotel Only stayed on...	Negative	Negative	Negative	Negative	Negative
5901	Grotty entrance hallway and reception doublin...	Negative	Negative	Negative	Negative	Negative
5902	The wifi was terrible Almost non existent I c...	Average	Positive	Positive	Positive	Positive
5903	The public space and reception are limited an...	Average	Positive	Negative	Negative	Negative
5904	The hotel was fine The room was clean but qui...	Average	Positive	Average	Average	Average

5905 rows × 6 columns

Figure 12. Final prediction result

```
In [ ]: required_df['Hotel_Name'].value_counts()

Out[129]: Britannia International Hotel Canary Wharf          724
          Hilton London Metropole                             288
          Grand Royale London Hyde Park                       264
          Holiday Inn London Kensington                       235
          Strand Palace Hotel                                 227
                                                              ...
          Hotel DO Pla a Reial G L                              1
          Hotel Square                                          1
          Relais Christine                                      1
          ADI Doria Grand Hotel                                 1
          Prince de Galles a Luxury Collection hotel Paris      1
          Name: Hotel_Name, Length: 1445, dtype: int64

In [ ]: britannia_hotel = required_df[required_df['Hotel_Name']=='Britannia International Hotel Canary Wharf']
```

Figure 13. List of hotels in our dataset

```
In [ ]: Britannia_hotel_Predictions
```

Out[158]:

	Overall_Review	purified_reviews	Melded BiGru and Fasttext
0	Single room is half a room with a full view t...	single room half room full view wall not reall...	Negative
1	Dated poorly maintained bedrooms need complet...	dated poorly maintained bedroom need complete ...	Negative
2	very noisy place couldn t sleep the whole nig...	noisy place sleep whole night expensive breakf...	Negative
3	Very unpleasant experience due to the lack of...	unpleasant experience due lack reactivity rega...	Negative
4	Staff were terrible no smiles were so abrupt ...	staff terrible smile abrupt guidance poor char...	Negative
...	...	...	...
719	Noisy fan in bedroom ceiling above bed You ca...	noisy fan bedroom ceiling bed cannot turn stop...	Average
720	No staff on enquiry desk Surely in the 21st c...	staff enquiry desk surely st century wifi free...	Average
721	Spondless health club staff is rudeNo Positive	spondless health club staff rudeno positive	Negative
722	An extra walk now from the tube station and w...	extra walk tube station workman nearby great b...	Average
723	Surly indifferent staff They seem to hate the...	surly indifferent staff seem hate job convenient	Average

724 rows × 3 columns

```
In [ ]: Britannia_hotel_Predictions["Melded BiGru and Fasttext"].value_counts()

Out[161]: Negative    478
          Average     201
          Positive     45
          Name: Melded BiGru and Fasttext, dtype: int64
```

Figure 14. Predictions for the Britannia Hotel

FINDINGS AND DISCUSSION

MODEL PERFORMANCE ANALYSIS

The proposed Bidirectional GRU model with FastText embeddings demonstrated superior performance, achieving an overall accuracy of 84.31% on the test dataset. This represents a significant improvement over baseline models, with paired t-tests confirming statistical significance ($p < 0.01$) compared to traditional approaches such as RF (57.32%) and SVM (47.23%). The integration of FastText embeddings proved particularly effective in handling the nuanced language of hotel reviews, capturing subtle semantic variations and context-dependent sentiments. The model showed

consistent performance across different review lengths and writing styles, with particularly strong performance in identifying extreme sentiments (positive and negative) compared to neutral reviews.

Feature Importance Analysis

Analysis of component contributions revealed that the FastText embedding layer played a crucial role in the model's success, accounting for approximately 40% of the performance improvement. The data purification pipeline significantly enhanced model accuracy by effectively handling informal language, emoticons, and abbreviated text common in user reviews. Our experiments showed that cleaned data improved classification accuracy by 12.3% compared to raw input. The incorporation of location information proved valuable, with geospatial features contributing to a 5.2% improvement in prediction accuracy, particularly for reviews where cultural context influenced sentiment expression.

Practical Implications

The developed model offers substantial practical value for the hotel industry, enabling real-time sentiment monitoring and location-specific trend analysis. Hotels can leverage this system to identify service areas requiring immediate attention and track sentiment patterns across different geographical locations. The model's scalability has been validated for processing up to 100,000 reviews per hour on standard cloud infrastructure, making it suitable for large-scale deployment. However, implementation challenges include the need for regular model retraining to adapt to evolving language patterns and the computational resources required for processing large volumes of reviews in real-time.

Limitations and Future Work

Despite the model's strong performance, several limitations warrant consideration. The current dataset, while comprehensive, is geographically limited to European hotels, potentially affecting generalizability to other regions. The computational requirements for model training and inference, particularly the memory demands of the FastText embedding layer, may pose challenges for deployment in resource-constrained environments. Future work should focus on extending the model to multilingual reviews, incorporating temporal analysis for trend prediction, and developing lighter model variants for edge deployment. Additionally, exploring the integration of aspect-based sentiment analysis could provide more granular insights into specific hotel service components.

Conclusion

This research presents significant contributions to sentiment analysis by developing a novel approach combining a fine-tuned bidirectional GRU architecture with FastText embeddings for location-based sentiment analysis.

This study's core contribution lies in its integration of FastText embeddings with a fine-tuned Bidirectional GRU architecture, optimized specifically for location-based sentiment analysis in the hospitality domain. Unlike transformer-based models that demand high computational resources, our approach offers a balanced tradeoff between performance and efficiency, making it well-suited for large-scale, real-time deployment. Moreover, the incorporation of location metadata enables a more granular understanding of customer sentiment across regions – an aspect often overlooked in sentiment analysis literature.

Our work advances the state-of-the-art in several key aspects. First, we demonstrate the superiority of our hybrid approach, achieving an accuracy of 84.31% on hotel review classification, substantially outperforming traditional machine learning methods and existing deep learning architectures. This improvement is particularly noteworthy given the complexity of sentiment analysis in the hospitality domain, where reviews often contain nuanced expressions and location-specific contexts. Second, we introduce an innovative data purification pipeline that effectively handles user-generated content

challenges, including informal language, emoticons, and geographical variations. This preprocessing framework and our architectural innovations provide a robust foundation for processing real-world review data. The integration of location-specific features with sentiment analysis represents a novel approach that captures the geographical context of customer opinions, enabling more nuanced and accurate sentiment predictions. Third, our comprehensive evaluation demonstrates the practical applicability of the model in real-world scenarios.

The model's ability to process large volumes of reviews while maintaining high accuracy makes it particularly valuable for the hospitality industry. The scalability and robustness of our approach enable real-time sentiment monitoring and location-based trend analysis, providing actionable insights for hotel management and customer experience improvement. Furthermore, this research opens new avenues for future work in sentiment analysis. The success of our location-aware approach suggests potential applications in other domains where geographical context influences sentiment expression. Future research directions include extending the model to multilingual scenarios, incorporating temporal analysis for trend prediction, and developing more efficient architectures for resource-constrained environments. In conclusion, this work not only advances the technical capabilities of sentiment analysis systems but also provides practical tools for the hospitality industry to better understand and respond to customer feedback. The demonstrated improvements in accuracy, coupled with the model's ability to handle real-world complexities, represent significant steps forward in the field of sentiment analysis and its applications in location-based services.

REFERENCES

Afzaal, M., Usman, M., & Fong, A. (2019). Tourism mobile app with aspect-based sentiment classification framework for tourist reviews. *IEEE Transactions on Consumer Electronics*, *65*(2), 233-242. https://doi.org/10.1109/TCE.2019.2908944

Badri, N., Kboubi, F., & Chaibi, A. H. (2022). Combining FastText and Glove word embedding for offensive and hate speech text detection. *Procedia Computer Science*, *207*, 769-778. https://doi.org/10.1016/j.procs.2022.09.132

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146. https://doi.org/10.1162/tacl_a_00051

Chang, V., Liu, L., Xu, Q., Li, T., & Hsu, C.-H. (2023). An improved model for sentiment analysis on luxury hotel review. *Expert Systems*, *40*(2), e12580. https://doi.org/10.1111/exsy.12580

Cheng, Y., Sun, H., Chen, H., Li, M., Cai, Y., Cai, Z., & Huang, J. (2021). Sentiment analysis using multi-head attention capsules with multi-channel CNN and bidirectional GRU. *IEEE Access*, *9*, 60383-60395. https://doi.org/10.1109/ACCESS.2021.3073988

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, *4*, 357-370. https://doi.org/10.1162/tacl_a_00104

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1724-1734). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1179

Farisi, A. A., Sibaroni, Y., & Al Faraby, S. (2019, March). Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier. *Journal of Physics: Conference Series, 1192*, 012024. https://doi.org/10.1088/1742-6596/1192/1/012024

Gualtieri, J. A., & Chettri, S. (2000, July). Support vector machines for classification of hyperspectral data. *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing, Honolulu, HI, USA,* 813-815. https://doi.org/10.1109/IGARSS.2000.861712

Haripriya, A., Kumari, S., & Babu, C. N. (2018, September). Location based real-time sentiment analysis of top trending event using hybrid approach. *Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Bangalore, India,* 1052-1057. https://doi.org/10.1109/ICACCI.2018.8554457

Harish, B. S., Kumar, K., & Darshan, H. K. (2019). Sentiment analysis on IMDb movie reviews using hybrid feature extraction method. *International Journal of Interactive Multimedia and Artificial Intelligence*, *5*(5), 109-114. https://doi.org/10.9781/ijimai.2018.12.005

Islam, K. I., Islam, M. S., & Amin, M. R. (2020, December). Sentiment analysis in Bengali via transfer learning using multilingual BERT. *Proceedings of the 23rd International Conference on Computer and Information Technology, Dhaka, Bangladesh,* 1-5. https://doi.org/10.1109/ICCIT51783.2020.9392653

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 427-431). Association for Computational Linguistics. https://doi.org/10.18653/v1/E17-2068

Kadli, P., & Vidyavathi, B. M. (2021). Deep-learned cross-domain sentiment classification using integrated polarity score pattern embedding on tri model attention network. *Indian Journal of Computer Science and Engineering*, *12*(6), 1910-1924. https://doi.org/10.21817/indjcse/2021/v12i6/211206190

Kaur, S., & Maini, R. (2020). Sentiment analysis on twitter data using machine learning. *Journal of Xidian University*, *14*(12), 370-384. https://doi.org/10.37896/jxu14.12/039

Khedkar, S., & Shinde, S. (2020). Deep learning and ensemble approach for praise or complaint classification. *Procedia Computer Science*, *167*, 449-458. https://doi.org/10.1016/j.procs.2020.03.254

Laoh, E., Surjandari, I., & Prabaningtyas, N. I. (2019, July). Enhancing hospitality sentiment reviews analysis performance using SVM N-grams method. *Proceedings of the 16th International Conference on Service Systems and Service Management, Shenzhen, China,* 1-5. https://doi.org/10.1109/ICSSSM.2019.8887662

Li, J., Chiu, B., Shang, S., & Shao, L. (2020). Neural text segmentation and its application to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, *34*(2), 828-842. https://doi.org/10.1109/TKDE.2020.2983360

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, *337*, 325-338. https://doi.org/10.1016/j.neucom.2019.01.078

Ma, Y., Xiang, Z., Du, Q., & Fan, W. (2018). Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep leaning. *International Journal of Hospitality Management*, *71*, 120-131. https://doi.org/10.1016/j.ijhm.2017.12.008

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. https://doi.org/10.48550/arXiv.1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Proceedings of the 27th International Conference on Neural Information Processing Systems* (pp. 3111-3119). Curran Associates Inc. https://doi.org/10.48550/arXiv.1310.4546

Mishev, K., Gjorgjevikj, A., Stojanov, R., Mishkovski, I., Vodenska, I., Chitkushev, L., & Trajanov, D. (2019). Performance evaluation of word and sentence embeddings for finance headlines sentiment analysis. In S. Gievska, & G. Madjarov (Eds.), *ICT innovations 2019: Big data processing and mining* (pp. 161-172). Springer. https://doi.org/10.1007/978-3-030-33110-8_14

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (pp. 79-86). Association for Computational Linguistics.

Penghua, Z., & Dingyi, Z. (2019). Bidirectional-GRU based on attention mechanism for aspect-level sentiment analysis. *Proceedings of the 2019 11th international conference on machine learning and computing* (pp. 86-90). Association for Computational Linguistics. https://doi.org/10.1145/3318299.3318368

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532-1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers), 2227–2237. https://doi.org/10.18653/v1/N18-1202

Puh, K., & Bagić Babac, M. (2023). Predicting sentiment and rating of tourist reviews using machine learning. *Journal of Hospitality and Tourism Insights*, *6*(3), 1188-1204. https://doi.org/10.1108/JHTI-02-2022-0078

Rana, R. (2016). *Gated recurrent unit (GRU) for emotion classification from noisy speech*. PsyArXiv. https://doi.org/10.48550/arXiv.1612.07778

Serna, A., Gerrikagoitia, J. K., & Bernabé, U. (2016). Discovery and classification of the underlying emotions in the User Generated Content (UGC). In A. Inversini, & R. Schegg (Eds.), *Information and communication technologies in tourism* (pp. 225-237). Springer. https://doi.org/10.1007/978-3-319-28231-2_17

Sivakumar, S., & Rajalakshmi, R. (2019). Comparative evaluation of various feature weighting methods on movie reviews. In H. Behera, J. Nayak, B. Naik, & A. Abraham (Eds.), *Computational intelligence in data mining* (pp. 721-730). Springer. https://doi.org/10.1007/978-981-10-8055-5_64

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 417-424). Association for Computing Machinery. https://doi.org/10.3115/1073083.1073153

Wankhede, R., & Thakare, A. N. (2017, April). Design approach for accuracy in movies reviews using sentiment analysis. *Proceedings of the International Conference of Electronics, Communication and Aerospace Technology, Coimbatore, India,* 6-11. https://doi.org/10.1109/ICECA.2017.8203652

Xiao, Z., & Liang, P. (2016). Chinese sentiment analysis using bidirectional LSTM with word embedding. In X. Sun, A. Liu, H. C. Chao, & E. Bertino (Eds.), *Cloud computing and security* (pp. 601-610). Springer. https://doi.org/10.1007/978-3-319-48674-1_53

Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on BiLSTM. *IEEE Access*, *7*, 51522-51532. https://doi.org/10.1109/ACCESS.2019.2909919

Yang, Y., & Liu, X. (1999, August). A re-examination of text categorization methods. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 42-49). Association for Computing Machinery. https://doi.org/10.1145/312624.312647

Yao, J., Wang, H., & Yin, P. (2011). Sentiment feature identification from Chinese online reviews. In H. Tan, & M. Zhou (Eds.), *Advances in information technology and education* (pp. 315-322). Springer. https://doi.org/10.1007/978-3-642-22418-8_44

Yaqub, U., Sharma, N., Pabreja, R., Chun, S. A., Atluri, V., & Vaidya, J. (2020). Location-based sentiment analyses and visualization of Twitter election data. *Digital Government: Research and Practice*, *1*(2), Article 14. https://doi.org/10.1145/3339909

APPENDIX: DICTIONARY OF CONTRACTIONS, EMOTICONS, AND ABBREVIATIONS USED FOR NORMALIZATION

Short form/contraction	Expanded version	Category
ain't	am not / are not	Contraction
can't	cannot	Contraction
won't	will not	Contraction
shouldn't	should not	Contraction
wouldn't	would not	Contraction
doesn't	does not	Contraction
doesn't	does not	Contraction
didn't	did not	Contraction
doesn't	does not	Contraction

Short form/contraction	Expanded version	Category
isn't	is not	Contraction
aren't	are not	Contraction
I've	I have	Contraction
you've	you have	Contraction
they've	they have	Contraction
it's	it is / it has	Contraction
I'd	I would / I had	Contraction
you'd	you would / you had	Contraction
we'd	we would / we had	Contraction
I'm	I am	Contraction
you're	you are	Contraction
we're	we are	Contraction
they're	they are	Contraction
I'll	I will	Contraction
you'll	you will	Contraction
we'll	we will	Contraction
there'd	there would have	Contraction
could've	could have	Contraction
might've	might have	Contraction
would've	would have	Contraction
doesn't	does not	Contraction
could not	could not	Contraction
should not	should not	Contraction
ain't	am not / are not	Contraction
can't	cannot	Contraction
mightn't	might not	Contraction
aren't	are not	Contraction
I'll	I will	Contraction
would've	would have	Contraction
:-(	sad	Emoticon
:(	sad	Emoticon
:c	sad	Emoticon
:<	sad	Emoticon
:)	happy	Emoticon
:-)	happy	Emoticon
:-]	happy	Emoticon
:-D	happy	Emoticon
:D	happy	Emoticon
:o	surprised	Emoticon
:P	playful, teasing	Emoticon
:-P	playful, teasing	Emoticon
:/	unsure, awkward	Emoticon
<3	love	Emoticon
:-		neutral, indifferent
O:-)	angel, good	Emoticon
LULZ	laugh out loud (slang)	Abbreviation
LMAO	laughing my ass off	Abbreviation
LOL	laugh out loud	Abbreviation

Short form/contraction	Expanded version	Category
BRB	be right back	Abbreviation
ROFL	rolling on the floor laughing	Abbreviation
TTYL	talk to you later	Abbreviation
SMH	shaking my head	Abbreviation
TBH	to be honest	Abbreviation
IDK	I don't know	Abbreviation
BFF	best friends forever	Abbreviation
DM	direct message	Abbreviation
FOMO	fear of missing out	Abbreviation
YOLO	you only live once	Abbreviation
IMO	in my opinion	Abbreviation
IMHO	in my humble opinion	Abbreviation
WTF	what the heck (or heck)	Abbreviation
TL;DR	too long, didn't read	Abbreviation
ASAP	as soon as possible	Abbreviation
TMI	too much information	Abbreviation
AF	as heck (intensifier)	Abbreviation
GG	good game	Abbreviation
BRB	be right back	Abbreviation
LDR	long distance relationship	Abbreviation
121	one to one	Abbreviation
A/S/L	age, sex, location	Abbreviation
ADN	any day now	Abbreviation
GA	go ahead	Abbreviation
GAL	get a life	Abbreviation
LLTA	lots and lots of thunderous applause	Abbreviation

AUTHORS

Ms Akshatha Shetty is a committed educator and accomplished soft skill trainer, with over 10 years of experience in academia. She is an Assistant Professor in the PG Department of Computer Applications at St. Agnes College (Autonomous), Mangalore. Her dedication to fostering learning and development extends beyond the classroom.

Dr. Manjaiah D.H., senior professor and chairman at Mangalore University, holds a PhD in Advanced Network Computing Paradigm. His expertise includes Routing Algorithms, Big Data Analytics, Recommendation Systems, 4G Handoff Mechanisms, and Digital Forensics. He has led projects funded by UGC, DST-SERB, and DST-VGST. With 153+ publications, his research spans Mobile Ad-Hoc Networks, IPv4/IPv6 mobility, AI, Cloud Virtualization, and Cybersecurity.

Ms Praveena Kumari M. K., a research scholar in the Department of PG Studies and Research in Computer Science at Mangalore University, MangalaGangotri, Mangalore, India, holds an MCA and an MPhil in Computer Science. She currently serves at the NMAM Institute of Technology, Nitte (Deemed to be University). Her research interests include Machine Learning, Deep Learning, Big Data Analytics, Natural Language Processing, Advanced Databases, and Generative AI.