Aspect-Based Sentiment Analysis and Topic Modeling of Hotel Reviews using BERT and LDA

Konstantinos Zafeiropoulos

AM: 20390293, ice20390293@uniwa.gr

University of West Attica - Department of Informatics and Computer Engineering

Class: Natural Language Processing and The Semantic Web

Teachers: P. Tselenti

Abstract

Our study presents a hybrid approach combining BERT-based sentiment analysis and Latent Dirichlet Allocation (LDA) for topic modeling of hotel reviews. By combining BERT-based sentiment analysis and LDA for topic modeling, our study provides a comprehensive framework for analyzing hotel reviews. This approach empowers stakeholders in the hospitality domain with actionable insights gleaned from sentiment trends, prevalent themes, and customer feedback, facilitating informed decision-making processes.

Initially, we meticulously prepare the dataset, "tripadvisor_hotel_reviews.csv," [1] by conducting preliminary inspections to understand its structure and visualize rating distributions. Text data undergoes comprehensive cleaning and preprocessing, including tokenization, stopwords removal, and visualization of token distributions to enhance data quality. Further preprocessing involves labeling ratings, visualizing distributions, identifying common words, and tokenizing reviews using BERT. This exploration offers insights into prevalent sentiments and common terms, priming the dataset for model development and analysis.

We then employ a BERT-based sentiment analysis pipeline, performing error analysis, model evaluation, and prediction aggregation to assess performance comprehensively. Simultaneously, LDA is applied for topic modeling, revealing latent themes and sentiments embedded in hotel reviews. Our approach includes defining a configuration class, splitting the dataset, encoding text data using BERT's tokenizer and training the model over multiple epochs. Evaluation metrics ensure robust model performance, with model checkpoints saved and the best model selected for final evaluation.

Keywords: Aspect-based sentiment analysis, Hotel Reviews, BERT, LDA, Preprocessing, Topic Modeling, Data Cleaning, Model Evaluation.

1. Introduction

The advancement of today's digital era has facilitated the booking of various travel necessities, including train tickets, flight tickets, tour bookings, hotel reservations, restaurant reservations, and similar services through online travel agency websites [2]. These websites also offer features for users to provide reviews on the services they have used, helping other users make informed decisions based on the feedback of previous customers. The results of these reviews can also serve as valuable data for service providers to assess the experiences and feedback from their customers, a process known as sentiment analysis in the field of information technology.

Hotel reviews have become particularly significant in sentiment analysis, as they offer insights into various aspects of hotels, aiding travelers in making well-informed decisions when selecting accommodations for their journeys [3]. Sentiment analysis, a computational method, automatically extracts customer opinions regarding a product [4]. Its evolution has led to aspect-based sentiment analysis (ABSA), a technique capable of assessing sentiments on predefined aspects. ABSA can deduce sentiment polarity based on aspect categories or target entities mentioned in the text. A prominent application of ABSA is in analyzing hotel review ratings [2, 5].

To align with the objectives of this study, latent Dirichlet allocation (LDA) [5] stands out as a prominent unsupervised learning method in topic modeling. It aids in identifying the underlying topics within a document by uncovering hidden topics. Prior hotel review research [2] has utilized LDA to extract aspect and opinion terms for aspect-based sentiment analysis (ABSA). However, similar to earlier studies [5], this research [2] faces limitations when applied to complex review cases containing multiple sentences. Consequently, inaccuracies may arise in aspect acquisition and opinion term extraction, impacting word similarity measurements.

The process of word similarity extraction, crucial for determining the similarity between aspect terms and categories, typically relies on semantic similarity [6]. Semantic similarity, based on tokens, plays a pivotal role in this process, especially in large-scale datasets with multiple sentences. Thus, there is a need to enhance word similarity procedures to ensure accuracy.

Recent studies [2] have explored innovative token-based text extraction techniques, such as BERT. BERT has emerged as a powerful tool capable of automatically processing large-scale data. In the word similarity process, BERT represents a significant advancement. However, further development is necessary to address potential issues, such as close word similarity values among different category classes, which can lead to errors in the categorization process [6].

This article aims to explore the integration of Latent Dirichlet Allocation and BERT for enhanced word similarity in Aspect-Based Sentiment Analysis of hotel reviews. By leveraging the strengths of both methodologies, we seek to overcome existing challenges and advance the accuracy and effectiveness of sentiment analysis in the hospitality domain.

2. State of the Art

2.1. Data Pre-Processing

The text extraction pre-processing stage involves transforming raw data into a format suitable for further analysis. This stage typically includes several steps: 1) case folding, 2) filtering, 3) normalization, 4) removing stop words, 5) stemming, and 6) tokenizing [7].

One approach in machine learning is the lexicon-based method, which relies on a predefined list of words derived from a corpus. The algorithm searches for these words, assigns them appropriate weights, and draws conclusions based on this analysis. Many methods and pipelines for processing positive and negative sentiment in NLP exist, and some of the most popular include the following methodology outlined in a referenced study [8]:

1. Tokenization: This involves converting a sentence into smaller units, or tokens.
2. Stopwords Removal: Stopwords, which do not carry significant meaning, are removed to streamline the text. There are four main approaches to this:

    - Pre-compiled List: Removing stopwords from a predefined list.

- Zipf's Law Method: Removing tokens with high term frequency (TF) values and words that appear only once, as well as tokens with low inverse document frequency.
- Mutual Information Method.
- Term Based Random Sampling.

3. Stemming: This process standardizes tokens to their root forms, so variations like 'waved', 'wave', and 'Wave' are considered the same. Several stemmers are recommended:

- Porter Stemmer: Uses a truncating method.
- N-Gram Stemmer: Statistical approach.
- Krovetz Stemmer: A more complex approach.

This pipeline is suggested for general text processing, but sentiment analysis, especially with data from micro-blogging sites like Facebook and Twitter, requires more sophisticated token configuration to handle the nuances of such platforms.

Article [9] provides an integrated approach to text preprocessing, emphasizing techniques to create cleaner text for better decision-making. Particularly on the internet, where noise like HTML tags and emojis is prevalent, effective preprocessing is crucial. The authors recommend not removing hashtags since they convey significant information. They also suggest using classifiers with higher accuracy and lower resource demands for large-scale data.

The proposed preprocessing pipeline includes:

1. Converting text to lowercase.
2. Removing @mentions.
3. Removing URLs.
4. Removing punctuation from non-numeric tokens.
5. Removing hashtags and saving them in a separate column.
6. Removing whitespace.
7. Tokenization.
8. Removing encoded text.
9. Removing stopwords.
10. Stemming.

In addition, the authors suggest three methods post-stemming: Stemming, Lemmatization, and Spelling Correction. While previous methodologies did not use lemmatization, it is a powerful tool that converts words to their present tense form, such as changing "ate" to "eat." This leads to more accurate word identification and reduced memory usage. In our study we will focus using these methodologies as it is seen later on the article.

2.2. LDA

The Latent Dirichlet Allocation (LDA) method is extensively utilized in topic modeling. It has proven to be more effective than Latent Semantic Analysis (LSA) and Probabilistic Latent Semantic Analysis (PLSA). The equation for the LDA method can be defined as:

$$p(w,z/\alpha,\beta)=p(w/\alpha,\beta)p(z/\alpha)$$

where α is first model parameter, β is second model parameter, w is word target in document, z is topic in document, and p(z/α) is topic z probability [4]

2.3. BERT

Bidirectional Encoder Representations from Transformers (BERT) is a significant advancement introduced by Google AI Language researchers. BERT utilizes Transformers, a deep learning model that links each output element to every input element, creating weights dynamically based on these connections. It is designed to pre-train deep bidirectional representations from unlabeled text by conditioning on both left and right contexts simultaneously. Consequently, the pre-trained BERT model can be fine-tuned with just an additional output layer, achieving state-of-the-art performance across various NLP tasks. BERT has set new benchmarks in eleven NLP tasks, including sentence-level sentiment classification [10].

The following are sentiment analysis steps using BERT method [11]:

1. Data pre-processing
2. Splitting data train and test
3. Tokenization, including token, segment, and
position embedding
4. Encoding
5. Set up model
6. Set up BERT pre-trained Model
7. Create data loaders
8. Set up optimizer and scheduler
9. Define performance matrix
10. Evaluation

The Robustly Optimized BERT Pretraining Approach (RoBERTa) is an enhanced version of BERT. It modifies key hyperparameters, removes the next sentence pretraining objective, and utilizes much larger mini-batches and learning rates. These adjustments further improve the model's performance [10].

2.4. RNN

Recurrent Neural Networks (RNNs) are a type of neural network designed to read and process sequences of data, such as documents split into sentences or characters known as n-grams. RNNs retain the sequence information of the data and utilize patterns to make predictions. They incorporate feedback loops that help in processing sequential data, allowing information to be shared across different nodes and predictions to be made based on accumulated information. This mechanism is often referred to as the network's "memory." The loops enable RNNs to share information and process input data sequences effectively. By converting an independent variable into a dependent variable for the next layer, RNNs can handle sequential input data. Commonly used RNN architectures for Toxic Comments Classification include LSTM (Long-Short-Term Memory Network), bidirectional LSTM, bidirectional GRU (Gated Recurrent Unit), and bidirectional GRU with an Attention Layer [10].

2.5. CNN

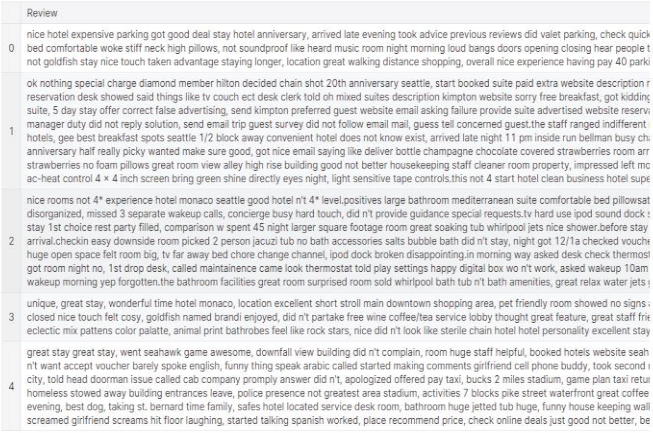
Convolutional Neural Networks (CNNs) are a crucial neural network model in the deep learning domain. They have achieved notable success in various fields, particularly in computer vision, and have also made significant strides in natural language processing. CNNs consist of three primary types of layers: convolutional layers, pooling layers, and fully connected layers [10]. The convolutional layer extracts features from the input data, producing a large volume of data that can be challenging for training. To manage this, the pooling layer compresses the data from the convolutional layer by reducing the size of the feature map. There are two main types of pooling: max pooling, which selects the maximum value from each patch of the feature map, and average pooling, which calculates the average value. The result is a condensed version of the features detected from the input. Finally, the fully connected layer uses these features for prediction. Stacking these layers creates a CNN architecture capable of detecting specific feature combinations, while RNNs focus on extracting sequential information. CNNs are particularly adept at handling character-level obfuscation of words [11].

3. Methodology

3.1. Data Acquisition – Presentation

The dataset used in this analysis was obtained from the TripAdvisor hotel reviews dataset, which consists of 20,491 reviews. This dataset was loaded into a Pandas DataFrame for subsequent analysis. To prepare the dataset for aspect-based sentiment analysis, several preprocessing steps were undertaken in total:

Essential libraries such as demoji, nltk, gensim, and others were installed and imported to facilitate the analysis. For instance, the demoji module was installed to handle emojis in the text data. To understand the distribution of ratings in the dataset, a bar chart was created. The bar chart illustrated that the majority of reviews were positive, with the highest frequency of ratings being 5, followed by 4. This visualization highlighted the overall sentiment trend in the dataset, indicating a generally positive reception of the hotels reviewed.



	Review
0	nice hotel expensive parking got good deal stay hotel anniversary, arrived late evening took advice previous reviews did valet parking, check quick bed comfortable woke stiff neck high pillows, not soundproof like heard music room night morning loud bangs doors opening closing hear people t not goldfish stay nice touch taken advantage staying longer, location great walking distance shopping, overall nice experience having pay 40 parki
1	ok nothing special charge diamond member hilton decided chain shot 20th anniversary seattle, start booked suite paid extra website description r reservation desk showed said things like tv couch ect desk clerk told oh mixed suites description kimpton website sorry free breakfast, got kidding suite, 5 day stay offer correct false advertising, send kimpton preferred guest website email asking failure provide suite advertised website reserv manager duty did not reply solution, send email trip guest survey did not follow email mail, guess tell concerned guest.the staff ranged indifferent hotels, gee best breakfast spots seattle 1/2 block away convenient hotel does not know exist, arrived late night 11 pm inside run bellman busy ch anniversary half really picky wanted make sure good, got nice email saying like deliver bottle champagne chocolate covered strawberries room arr strawberries no foam pillows great room view alley high rise building good not better housekeeping staff cleaner room property, impressed left mo ac-heat control 4 × 4 inch screen bring green shine directly eyes night, light sensitive tape controls.this not 4 start hotel clean business hotel supe
2	nice rooms not 4* experience hotel monaco seattle good hotel n't 4* level.positives large bathroom mediterranean suite comfortable bed pillowsat disorganized, missed 3 separate wakeup calls, concierge busy hard touch, did n't provide guidance special requests.tv hard use ipod sound dock stay 1st choice rest party filled, comparison w spent 45 night larger square footage room great soaking tub whirlpool jets nice shower.before stay arrival.checkin easy downside room picked 2 person jacuzi tub no bath accessories salts bubble bath did n't stay, night got 12/1a checked vouche huge open space felt room big, tv far away bed chore change channel, ipod dock broken disappointing.in morning way asked desk check thermost got room night no, 1st drop desk, called maintainence came look thermostat told play settings happy digital box wo n't work, asked wakeup 10am wakeup morning yep forgotten.the bathroom facilities great room surprised room sold whirlpool bath tub n't bath amenities, great relax water jets
3	unique, great stay, wonderful time hotel monaco, location excellent short stroll main downtown shopping area, pet friendly room showed no signs closed nice touch felt cosy, goldfish named brandi enjoyed, did n't partake free wine coffee/tea service lobby thought great feature, great staff frie eclectic mix pattens color palatte, animal print bathrobes feel like rock stars, nice did n't look like sterile chain hotel hotel personality excellent stay
4	great stay great stay, went seahawk game awesome, downfall view building did n't complain, room huge staff helpful, booked hotels website seah n't want accept voucher barely spoke english, funny thing speak arabic called started making comments girlfriend cell phone buddy, took second city, told head doorman issue called cab company promply answer did n't, apologized offered pay taxi, bucks 2 miles stadium, game plan taxi retu homeless stowed away building entrances leave, police presence not greatest area stadium, activities 7 blocks pike street waterfront great coffee evening, best dog, taking st. bernard time family, safes hotel located service desk room, bathroom huge jetted tub huge, funny house keeping wall screamed girlfriend screams hit floor laughing, started talking spanish worked, place recommend price, check online deals just good not better, be

Fig. 1. Example of data collection

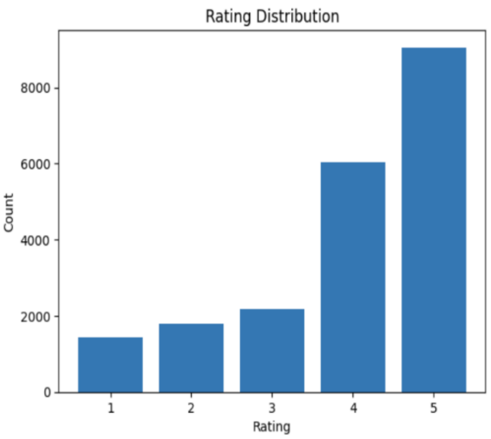
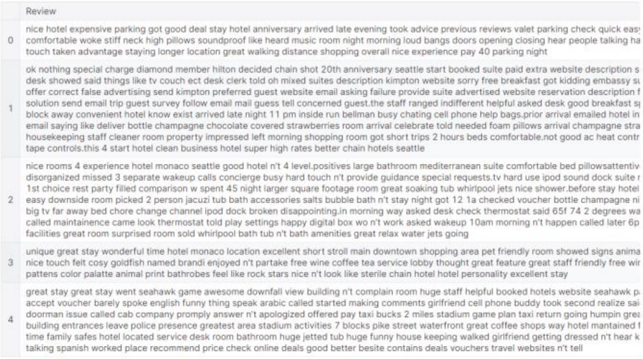


Fig. 2. Rating Distribution

3.2. Text Cleaning & Text Pre-Processing

More thoroughly, the text undergoes data clearing processes, which involve the removal of unwanted elements such as URLs, emojis, emoticons, and unicode normalization. URLs are removed using regular expressions (regex), emojis are removed using the demoji library, and unicode normalization is performed using the unicodedata module. After that, the text undergoes further pre-processing steps, including tokenization and stop words removal. Tokenization splits the text into individual tokens, such as words or punctuation marks, while stop words, which are common words that do not carry significant meaning (e.g., "the", "is", "and"), are removed from the text. All of the aforementioned functions are encapsulated within the TextProcessor class, making it a cohesive unit for text processing operations. Additionally, a separate tokenization function is defined to tokenize the processed text further. This function utilizes regular expressions to split the text into tokens based on alphanumeric characters and non-alphanumeric characters.



	Review
0	nice hotel expensive parking got good deal stay hotel anniversary arrived late evening took advice previous reviews valet parking check quick eas comfortable woke stiff neck high pillows soundproof like heard music room night morning loud bangs doors opening closing hear people talking ha touch taken advantage staying longer location great walking distance shopping overall nice experience pay 40 parking night
1	ok nothing special charge diamond member hilton decided chain shot 20th anniversary seattle start booked suite paid extra website description s desk showed said things like tv couch ect desk clerk told oh mixed suites description kimpton website sorry free breakfast got kidding embassy su offer correct false advertising send kimpton preferred guest website email asking failure provide suite advertised website reservation description f solution send email trip guest survey follow email mail guess tell concerned guest.the staff ranged indifferent helpful asked desk good breakfast sp block away convenient hotel know exist arrived late night 11 pm inside run bellman busy chating cell phone help bags.prior arrival emailed hotel in email saying like deliver bottle champagne chocolate covered strawberries room arrival celebrate told needed foam pillows arrival champagne stra housekeeping staff cleaner room property impressed left morning shopping room got short trips 2 hours beds comfortable.not good ac heat contr tape controls.this 4 start hotel clean business hotel super high rates better chain hotels seattle
2	nice rooms 4 experience hotel monaco seattle good hotel n't 4 level.positives large bathroom mediterranean suite comfortable bed pillowsattentiv disorganized missed 3 separate wakeup calls concierge busy hard touch n't provide guidance special requests.tv hard use ipod sound dock suite r 1st choice rest party filled comparison w spent 45 night larger square footage room great soaking tub whirlpool jets nice shower.before stay hotel easy downside room picked 2 person jacuzi tub bath accessories salts bubble bath n't stay night got 12 1a checked voucher bottle champagne ni big tv far away bed chore change channel ipod dock broken disappointing.in morning way asked desk check thermostat said 65f 74 2 degrees wa called maintainence came look thermostat told play settings happy digital box wo n't work asked wakeup 10am morning n't happen called later 6p facilities great room surprised room sold whirlpool bath tub n't bath amenities great relax water jets going
3	unique great stay wonderful time hotel monaco location excellent short stroll main downtown shopping area pet friendly room showed signs anima nice touch felt cosy goldfish named brandi enjoyed n't partake free wine coffee tea service lobby thought great feature great staff friendly free wir pattens color palatte animal print bathrobes feel like rock stars nice n't look like sterile chain hotel hotel personality excellent stay
4	great stay great stay went seahawk game awesome downfall view building n't complain room huge staff helpful booked hotels website seahawk p accept voucher barely spoke english funny thing speak arabic called started making comments girlfriend cell phone buddy took second realize sai doorman issue called cab company promply answer n't apologized offered pay taxi bucks 2 miles stadium game plan taxi return going humpin grea building entrances leave police presence greatest area stadium activities 7 blocks pike street waterfront great coffee shops way hotel mantained f time family safes hotel located service desk room bathroom huge jetted tub huge funny house keeping walked girlfriend getting dressed n't hear k talking spanish worked place recommend price check online deals good better besite contains deals vouchers travel websites n't tell

Fig. 3. Example of cleaned data

3.3. Text Pre-Processing & EDA

The process begins with labeling the ratings in a dataset containing hotel reviews. The ratings are categorized into three classes: 'negative', 'neutral', and 'positive'. Ratings of 1 and 2 are labeled as 'negative' (0), a rating of 3 is labeled as 'neutral' (1), and ratings of 4 and 5 are labeled as 'positive' (2). This categorization is done using the label_rating function. Subsequently, the label_name function translates these numerical labels into their corresponding descriptive names. These labeled columns are then added to the DataFrame: 'Rating_Label' for numerical labels and 'Label_Name' for descriptive labels. Following the labeling process, the distribution of ratings is visualized. A bar plot is created to show the count of each rating label, revealing that positive ratings are the most frequent, followed by negative and neutral ratings. This visualization helps in understanding the sentiment distribution within the dataset. To analyze the text data further, a word cloud is generated to identify the most common words in the reviews. This is achieved by combining all reviews into a single string and then using the WordCloud class to visualize the most frequent words, providing insights into common themes and terms used in the reviews.

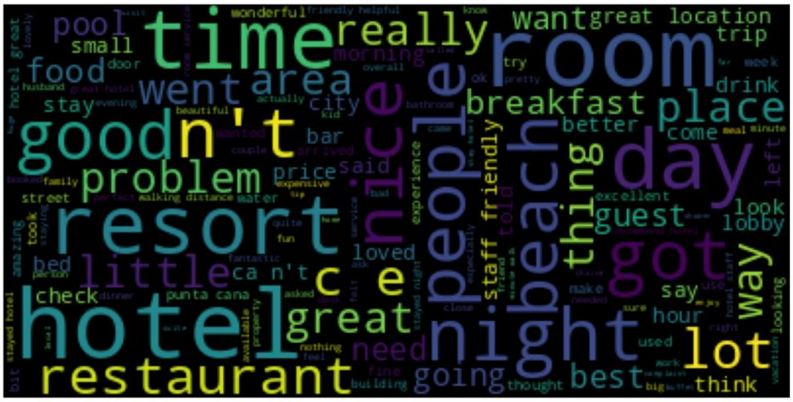


Fig. 4. Most frequent words - WorkCloud

For a more detailed text analysis, BERT tokenizer from the transformers library is used. Each review is tokenized, and the token lengths are calculated and added to the DataFrame as 'sent_bert_token_length'. Additionally, the character count of each review is computed and stored in the 'char_count' column. These features provide valuable information about the length and complexity of the reviews. A histogram is then plotted to show the distribution of a selected feature, such as 'sent_bert_token_length'. This histogram provides a visual representation of how the token lengths are distributed, which can be useful for understanding the variability and common patterns in the review lengths. Lastly, the most common words in the reviews are identified by splitting the reviews into individual words, aggregating them into a corpus, and counting their occurrences using the Counter class. A horizontal bar chart is then plotted to display the top 30 most frequent words, excluding common stop words. This chart helps in identifying key terms and common vocabulary used by reviewers, which can be indicative of prevalent themes and sentiments in the reviews.

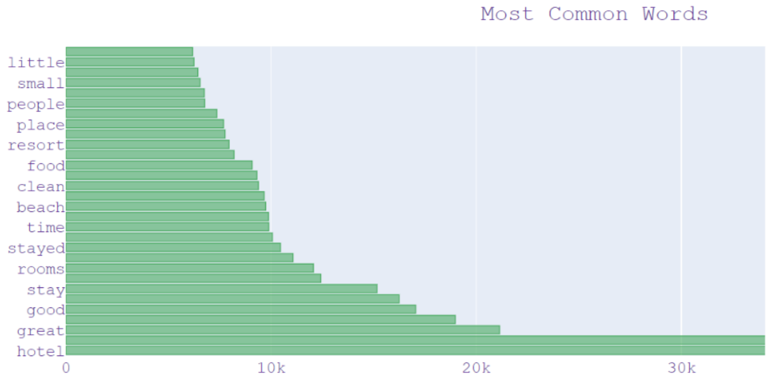


Fig. 5. Most common words

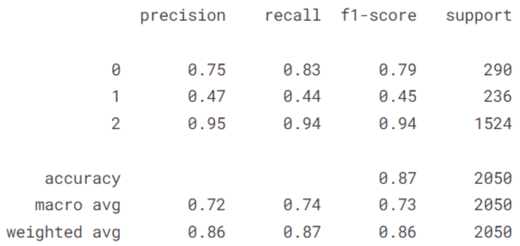
3.4. Modeling with BERT

We begin with the initialization of necessary libraries and configuration parameters. Libraries such as torch, transformers, and tqdm are imported, alongside utility libraries like os, random, and pathlib. A configuration class (Config) is defined to store parameters for the training process, including the seed value, device type, number of epochs, batch size, sequence length, learning rate, epsilon for the optimizer, pre-trained model name, test size, random state, and various tokenizer settings. These parameters are stored in a dictionary for easy access and saving. The random seeds for reproducibility are set using the random, numpy, and torch libraries. The dataset is split into training, validation, and test sets using train_test_split from the sklearn.model_selection module, ensuring that the splits are stratified based on the Rating_Label column to maintain the label distribution across sets.

The BERT tokenizer is initialized using the pre-trained model specified in the configuration. The text data from the training and validation sets is then encoded using the tokenizer's batch_encode_plus method, which adds special tokens, returns attention masks, pads sequences to the maximum length, and converts the text data into tensors. The encoded data is then stored in TensorDatasets, which include the input IDs, attention masks, and labels for both the training and validation sets. A BERT model for sequence classification is created using BertForSequenceClassification, with the specified pre-trained model and the number of labels set to three. The model is then moved to the appropriate device.

Data loaders are created for the training and validation datasets using DataLoader from torch.utils.data, with a RandomSampler for the training set and a SequentialSampler for the validation set. These data loaders ensure that the data is efficiently loaded in batches during training and evaluation. The optimizer and learning rate scheduler are initialized using AdamW and get_linear_schedule_with_warmup from the transformers library. The optimizer is set with the learning rate and epsilon values from the configuration, and the scheduler is configured to linearly warm up the learning rate over the training steps. Performance metrics functions are defined using f1_score from sklearn.metrics to evaluate the model's performance. The f1_score_func calculates the weighted F1 score, and accuracy_per_class prints the accuracy for each class.

Last but not least, the training process involves multiple epochs, with each epoch iterating over the training data loader. For each batch, the model is set to training mode, gradients are reset, inputs are moved to the device, and the model is run forward to calculate the loss. The loss is backpropagated, and the optimizer steps are performed, followed by a scheduler step to update the learning rate. The training loss is tracked and saved at the end of each epoch while the model's state is saved after each epoch. Evaluation on the validation set is performed, using the evaluate function, which sets the model to evaluation mode, disables gradient calculation, and processes the validation batches. The average validation loss, predictions, and true labels are calculated and returned while the F1 score for the validation set is computed and printed. The trained model's parameters and configuration are saved to a JSON file for future reference and the model is loaded for testing using the saved state from the epoch with the best performance. Finally, the classification report is generated using classification_report from sklearn.metrics, which provides precision, recall, F1-score, and support for each class, as well as overall accuracy and weighted averages:

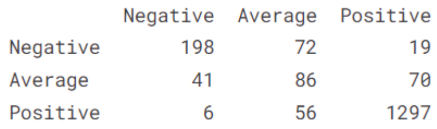


```
              precision    recall  f1-score   support

           0       0.75      0.83      0.79       290
           1       0.47      0.44      0.45       236
           2       0.95      0.94      0.94      1524

    accuracy                           0.87      2050
   macro avg       0.72      0.74      0.73      2050
weighted avg       0.86      0.87      0.86      2050
```

Fig. 5. Result of ABSA

3.5. Error Analysis – BERT

Error analysis on the BERT model includes predictions on the validation and test sets, computing confusion matrices, and evaluating the model's performance using various metrics. The process begins with making predictions on the validation set. For each review in the validation dataset, the review text is tokenized using the BERT tokenizer's batch_encode_plus method. This method adds special tokens, returns attention masks, pads sequences to a specified length, and converts the text data into tensors. The input IDs and attention masks are then moved to the appropriate device (GPU or CPU). The model is set to evaluation mode, and predictions are made without gradient calculation to save memory and computation. The model outputs logits, which are detached from the computation graph and converted to numpy arrays. The predicted class label is determined by finding the index of the maximum logit value. This predicted label is appended to a list of final predictions. Once predictions for the entire validation set are made, they are added to the validation dataframe. A confusion matrix is then computed using the confusion_matrix function from sklearn.metrics, which compares the true labels with the predicted labels. This matrix summarizes model's performance by showing the counts of true positive, true negative, false positive, and false negative predictions for each class. The confusion matrix is converted into a dataframe for better visualization and printed.

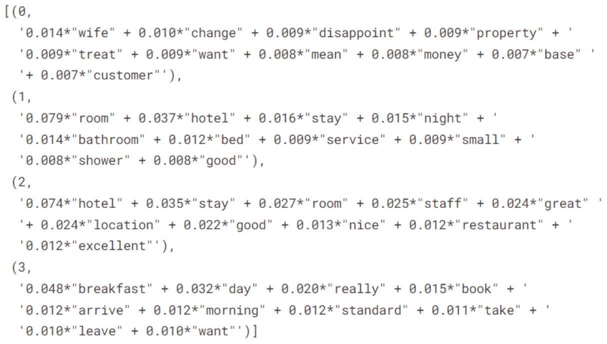


```
          Negative  Average  Positive
Negative       198       72        19
Average         41       86        70
Positive         6       56      1297
```

Fig. 6. Predictions for each class

3.6. Modeling with LDA

In the last chapter, we focus on modeling with Latent Dirichlet Allocation (LDA) to identify topics within a subset of hotel reviews. For this example, we use 5,000 reviews out of a total of 20,491. Initially, we confirm the dataset's dimensions to verify the number of reviews, which outputs the shape of the dataset as (20491, 8). We then select the first 5,000 reviews for analysis. Next, we tokenize and clean the sentences. This involves removing emails, newline characters, and single quotes, and then tokenizing the text into individual words. We convert the reviews into a list and tokenize them, which outputs a tokenized version of the first review. To enhance the analysis, we build bigram and trigram models to capture common phrases. These models identify frequent two-word and three-word combinations, which are then incorporated into the tokenized text. We then define a function to remove stopwords, form bigrams and trigrams, and perform lemmatization. This function processes the text data by eliminating common but uninformative words, grouping frequent word pairs and triplets, and converting words to their base forms. The processed text data is then ready for further analysis. To create a dictionary and a corpus, we map each word to a unique ID and represent each document as a list of word frequency pairs. The dictionary serves as a reference for the unique IDs assigned to words, while the corpus consists of documents represented by these IDs and their corresponding frequencies. We then build the LDA model using the corpus and dictionary, specifying parameters such as the number of topics and the number of iterations. The model is trained to identify patterns in the word distributions and generate topics which we then print. The output shows the top words for each of the four topics, providing insight into the themes present in the reviews. Finally, we use pyLDAvis to create an interactive visualization of the topics. This tool provides an overview of the model and allows us to closely examine the words associated with each topic. The visualization helps in understanding the inter-topic distances and the distribution of words within each topic, facilitating a deeper insight into the thematic structure of the hotel reviews. Through this comprehensive analysis, we can better understand the common themes and sentiments expressed by the reviewers.



```
[(0,
  '0.014*"wife" + 0.010*"change" + 0.009*"disappoint" + 0.009*"property" + '
  '0.009*"treat" + 0.009*"want" + 0.008*"mean" + 0.008*"money" + 0.007*"base" '
  '+ 0.007*"customer"'),
 (1,
  '0.079*"room" + 0.037*"hotel" + 0.016*"stay" + 0.015*"night" + '
  '0.014*"bathroom" + 0.012*"bed" + 0.009*"service" + 0.009*"small" + '
  '0.008*"shower" + 0.008*"good"'),
 (2,
  '0.074*"hotel" + 0.035*"stay" + 0.027*"room" + 0.025*"staff" + 0.024*"great" '
  '+ 0.024*"location" + 0.022*"good" + 0.013*"nice" + 0.012*"restaurant" + '
  '0.012*"excellent"'),
 (3,
  '0.048*"breakfast" + 0.032*"day" + 0.020*"really" + 0.015*"book" + '
  '0.012*"arrive" + 0.012*"morning" + 0.012*"standard" + 0.011*"take" + '
  '0.010*"leave" + 0.010*"want"')]
```

Fig. 7. Topics and their most significant words

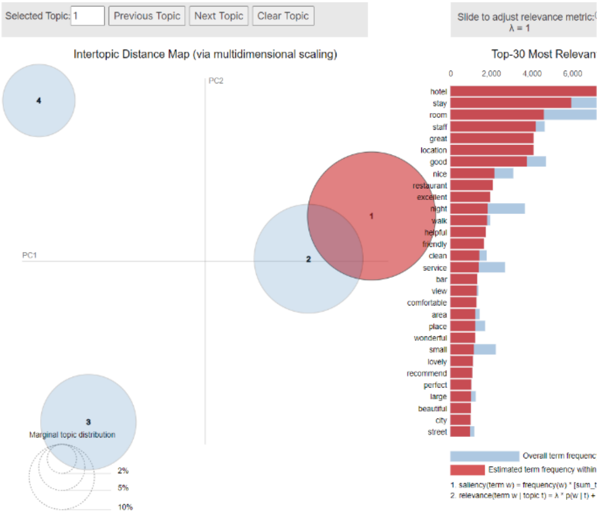


Fig. 8. pyLDAvis overview of Topic 1

4. Results and Discussion

4.1. BERT Model

Training Loss: Decreased across epochs, indicating that the model was learning.

Validation Loss and F1 Score:

- Epoch 1: Validation loss = 0.582, F1 Score = 0.824
- Epoch 2: Validation loss = 0.429, F1 Score = 0.853
- Epoch 3: Validation loss = 0.541, F1 Score = 0.865
- Epoch 4: Validation loss = 0.628, F1 Score = 0.866

The final F1 score indicates that the model performs well on the validation set, achieving a weighted F1 score of 0.866, which suggests good classification performance across different classes.

The classification report on the test set showed:

- Accuracy: 0.87
- Weighted Average F1-Score: 0.86

- Class-wise Performance:

  - Class 0 (Negative): Precision = 0.75, Recall = 0.83, F1-Score = 0.79
  - Class 1 (Neutral): Precision = 0.47, Recall = 0.44, F1-Score = 0.45
  - Class 2 (Positive): Precision = 0.95, Recall = 0.94, F1-Score = 0.94

4.2. LDA Model

The LDA model identified four topics, each with significant words:

- Topic 0: Focuses on customer dissatisfaction and monetary concerns, with words like "disappoint," "money," and "customer." This topic likely captures reviews that express negative experiences related to value for money or poor customer service.

- Topic 1: Pertains to hotel room conditions and amenities, with words like "room," "hotel," "stay," and "bed." This topic captures general reviews about the physical aspects of the hotel stay, including rooms and facilities.

- Topic 2: Highlights positive experiences, with words like "great," "location," "good," and "excellent." This topic is associated with positive feedback about the hotel, indicating satisfaction with location and service.

- Topic 3: Related to breakfast and daily routines, with words like "breakfast," "morning," and "arrive." This topic focuses on specific aspects of the stay related to food and morning activities, likely reflecting guest experiences with hotel breakfasts and check-in/check-out processes.

4.3. Discussion

The BERT model yielded a robust classifier with a high weighted F1 score, indicating effective performance in categorizing reviews into positive, neutral, and negative sentiments. The gradual reduction in training loss and the reasonable validation loss suggest that the model generalizes well without significant overfitting. On the other hand, the LDA model provided insightful clusters of topics prevalent in the hotel reviews. The identified topics align well with common themes in hotel reviews, such as room conditions, overall satisfaction, and specific amenities like breakfast. The differentiation between positive and negative experiences, as well as the focus on particular aspects like customer service and monetary value, demonstrates the model's capability to capture distinct themes from the textual data.

5. Future work and Suggestions

5.1. Continues Evolution of Languages

Every language possesses its own intricacies and evolves continuously. This evolution occurs through the addition of new grammatical rules and the introduction of new words into the lexicon. As a result, it is impossible to maintain data that is always current, and this challenge is likely to persist [10].

5.2. Adaptation of the Dictionary of Languages

Online communities are constantly evolving, altering their language use as they do so. New communities emerge while older ones decline or disappear. Each new community develops its own slang or modifies existing slang to suit its needs, incorporating new words into their unique "dictionary." Consequently, there is a necessity for a continuously adaptive method to effectively collect and stay updated with this dynamic data [10].

5.3. Enhancing Aspect Categorization Methods

Other methods for aspect categorization can be developed to improve the accuracy of extracting aspects and opinions from text. This is particularly important in cases involving: 1) implicit aspects, 2) implicit opinions, and 3) both implicit aspects and opinions. Additionally, it is essential to develop techniques for extracting aspect terms as word phrases, which can capture multiple categories of aspects within a single review [5, 6].

6. Conclusions

This study started with the acquisition and initial inspection of the TripAdvisor hotel reviews dataset, ensuring its integrity by loading it into a Pandas DataFrame and summarizing its structure. The preprocessing stage involved meticulous cleaning and normalization of the text data, which included removing unwanted elements and tokenizing the text, preparing it for further analysis. For sentiment analysis, reviews were labeled based on their ratings, and the BERT model was fine-tuned on this dataset to classify sentiments into negative, neutral, or positive categories effectively. The dataset was split into training, validation, and test sets to maintain balanced label distribution. Model’s performance was evaluated using metrics like precision, recall, and F1 score, with error analysis providing insights into its strengths and weaknesses. Topic modeling was then conducted on a subset of reviews to identify underlying themes using the LDA model. The text was further cleaned and transformed into bigrams and trigrams, enhancing the analysis. An interactive visualization of the topics was created, revealing deeper insights into the reviews' themes.

In conclusion, this study successfully converted raw textual data into structured information for sentiment and topic analysis. The analysis uncovered a predominantly positive sentiment trend and demonstrated the effective sentiment classification capabilities of the fine-tuned BERT model. Topic modeling provided deeper insights into common narratives within the reviews. This comprehensive approach offers valuable insights for decision-making in marketing and customer service improvement, highlighting both quantitative sentiment trends and qualitative themes.

References

[1] Alam, M. H., Ryu, W.-J., Lee, S., 2016. Joint multi-grain topic senti- ment: modeling semantic aspects for online reviews. Information Sci- ences 339, 206–223.
[2] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews", Procedia Comput. Sci., Vol. 179, pp. 728–735, 2021, doi: 10.1016/J.PROCS.2021.01.061.
[3] Ozyurt, B. and Akcayol, M.A., 2021. A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA. Expert Systems with Applications, 168, p.114231.
[4] B. S. Rintyarna, R. Sarno, and C. Fatichah, "Semantic features for optimizing supervised approach of sentiment analysis on product reviews", Computers, Vol. 8, No. 55, pp. 1-16, 2019, doi: 10.3390/computers8030055
[5] D. Khotimah and R. Sarno, "Sentiment Analysis of Hotel Aspect Using Probabilistic Latent Semantic Analysis, Word Embedding and LSTM", Int. J. Intell. Eng. Syst., Vol. 12, No. 4, pp. 275–290, 2019, doi: 10.22266/ijies2019.0831.26.
[6] S. Suhariyanto, R. Abdullah, R. Sarno, and C. Fatichah, "Aspect-Based Sentiment Analysis for Sentence Types with Implicit Aspect and Explicit Opinion in Restaurant Review Using Grammatical Rules, Hybrid Approach, and SentiCircle", Int. J. Intell. Eng. Syst., Vol. 14, No. 5, pp. 177–187, 2021, doi: 10.22266/ijies2021.1031.17.
[7] A. A. Farisi, Y. Sibaroni, and S. A. Faraby, "Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier", J. Phys. Conf. Ser., Vol. 1192, No. 1, 2019, doi: 10.1088/1742-6596/1192/1/012024
[8] Vijayarani, S., & Research Scholar, M. P. (n.d.). Preprocessing Techniques for Text Mining-An Overview.
[9] Pradha, S., Halgamuge, M. N., & Tran Quoc Vinh, N. (2019). Effective text data preprocessing technique for sentiment analysis in social media data. Proceedings of 2019 11th International Conference on Knowledge and Systems Engineering, KSE 2019. https://doi.org/10.1109/ KSE.2019.8919368
[10] Margaris, A. and Charalampidis, J., "Toxic Comment Classification". Department of Informatics and Computer Engineering, University of West Attica.
[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., Vol. 1, No. Mlm, pp. 4171–4186, 2019.