**Performing Leading NLP Methods Topic Modeling, Text Classification**

**and Sentiment Analysis on Hotel Reviews**

Mdakbar Sarkar, Robert Fajardo, Rajesh Pahari, Platini Dacheu

University of North Texas

CSCE 5290 : Natural Language Processing

Dr. Zeenat Tariq

November 29, 2022

**GitHub Link**

[**https://github.com/RobFaj/NLP\_Project2022**](https://github.com/RobFaj/NLP_Project2022)

**Video Link**

Final Project

**Topic**

Hotel ratings based on customer review and classification & sentimental analysis of review. The implementation will allow hotel owners/managers decide what areas they can improve or adjust to their environment, and leverage the data for insights into a 360 view of consumers and their sentiment.

**Motivation**

The goal is to find a good hotel and having an enjoyable vacation is everyone’s very basic expectation which will not be feasible without having customer feedback or review on the hotel. As we are living in a world where data plays an important role in the decision-making process for both customers and service/accommodation providers. We need a developed system based on customer reviews to make an unbiased rating of the hotel that not only helps the customer to choose the best one but also opens the opportunity to improve for the service provider. We need to have an effective system in place, we need to do the classification of review as well as sentimental analysis.

**Objectives**

The main objective of this project is to build a data centric application that not only helps the customer but also identifies key areas to improve from the provider’s perspective. After completion of the project, we will have a solid understanding of NLP and how we use the concept in the practical field.

**Significance**

You can’t manage what you can’t measure. With the advancement of technology and the rise of the internet, now the world becomes a global village. The recent explosion of digital data opens the door to a new way of thinking, working, and living. Information is the key but managing information is important as making a critical business decision. For example, if anyone wants to travel to a new place and doesn’t have any knowledge about the vicinity, then the plan for vacation may not materialize. In this case, customer reviews will play a big role to make the marketing strategy successful. At the same time, if we have millions of reviews without labeling and sentimental analysis, no one will go for the review of millions of customers. So, managing customer reviews in an efficient manner are necessary to promote, improve, and flourish unseen opportunities into fruition.

**Features**

We will use the following NLP features: tagging, summarization, and sentiment analysis. Within hotel ratings, we mainly focus on the sentimental analysis of customers - positive, negative, or neutral feedback which can be measured on a scale of 1 to 5 where 1 is low and 5 is excellent.

We will tag and classify the review with labels. We will use features such as cost, crime history, amenities, flexibility, parking, indoor facility, room space, air-conditioning, and neighborhood to help classify the hotels. In order to tag reviews effectively we assign words using text classification model. Then, we assign topic models to reviews to gain a better understanding of the review data and the most prevalent topics and their top 10 terms. Finally, we will use labeled classified data based on sentimental analysis to gauge the consumer sentiment and gain insights to consult providers in overall managerial decision making.

**Background**

Research or project topics are usually not formulated for the first time. Thus, the often build on previous works either in the same domain or some close domain. They can even be inspired by a totally different branch of science. Our topic does not escape this generality. To understand the state of the art of the problem, we have researched previous works accomplished in the area. This helped us more precisely fix the boundaries of our topics. Therefore, areas like sentiment analysis, product rating and review have been deeply explored to help us accomplish our task.

The reason topic modeling is useful is that it allows the user to not only explore what’s inside their corpus (documents) but also build new connections between topics they weren’t even aware of. LDA (Latent Dirichlet Allocation) is the most popular algorithm and we have utilized it in our project since it’s more adapted and we can find more literature and work on it. Another method is Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM). LDA assumes that each document (reviews in our case) consists of multiple topics and calculates the contribution of each topic to the document. GSDMM, on the other hand, is specifically aimed at detecting topics in smaller documents and assumes only one topic per document. Researches on PRTM (Ikegami, Kenshin; Ohsawa, Yukio 2018) showed that page rank topic model can infer an appropriate number of topics by clustering short sentences, and it was particularly effective when the sentences were covered by a small number of topics which is a benefit in our case since it can cover large number of topics.

Text classification can be described as a machine learning technique to classify the text into a specific category . Multinomial Naive Bayes, Random Forest, LSTM, CNN and combination of these methods can be the different way we can do text classifications. Unlike the typical CNN, which contains convolution operation and activation function, this paper constructs two text classification models called NA-CNN-LSTM and NACNN-COIF-LSTM by combining CNN without activation function and LSTM, and one of its variants COIF-LSTM (Luan, et. al 2019). Through comparative experiments, it is proved that the combination of CNN without activation function and LSTM or its variant has better performance (Luan, et. al 2019) . For our project, we have utilized LSTM due to It’s ability to outperform other methods provided more time and powerful hardware due to the utilization of back-propagation.

Sentiment analysis is a subfield of NLP that draws on approaches from information retrieval and computational linguistics to identify opinions expressed in text. It is considered a specific type of text mining (Han et al., 2011), and it has been called opinion mining. The main goal of sentiment analysis is to identify positive or negative overall attitudes or opinions toward a brand, product or service based on text comments (Liu, 2010). While the terms appraisal extraction or review mining have also been applied, they are not always completely accurate (Pang and Lee, 2008). Several machine learning and data mining algorithms have been used to detect sentiment (Khoo et al., 2012), and sentiment strength (Thelwall et al., 2010). Pang and Lee (2005) predicted star ratings of movie reviews based on a five-point sentiment scale instead of merely classifying the reviews as positive or negative. They employed a novel similarity measure with a meta-algorithm based on metric labeling and performed. several comparisons of pairs of reviews to identify when the first review was less positive than, more positive than or as positive as the second review. Even simple algorithms have been shown to work well with large data sets, as in the case of the naïve Bayesian approach (Wu and Kumar, 2008). Rutilo et al (2015) rated the hotel by the transformation of the positive percentage of its comments. We also referred to a paper on n enhanced feature extraction and classification model using BERT model and dilated convolutional Bi-LSTM model (Asghar , et. al, 2022) . A BERT-based CBRNN SA model was been proposed for sentence-level classification. The performance of the CBRNN model is evaluated using five statistical measures, such as accuracy, precision, f1-score, recall, and AUC. The obtained results were then compared with the most commonly used embedding models, such as glove and word2vec. The proposed model obtained significant improvement in f1-score 0.2%, accuracy 0.3% and AUC 0.4%. The experimental results concluded that the proposed CBRNN model is more efficient as compared to the other models. For our project we use a combination of Transformer and BERT Model. BERT learns contextual relation between words using transformer.

Our current project aims to utilize rules based approach in order to generate the hotel rating based on customer review and classification & sentimental analysis of review. Rule based approach is used by defining various rules for getting the opinion, created by tokenizing each sentence in every document and then testing each token, or word, for its presence.

**Model Interpretation**

**Topic Modeling**

Topic modeling is an unsupervised approach of extracting the topics by detecting the pattern like clustering. We can extract different topics from documents. This is done by extracting the pattern of word cluster and frequency of words in the documents. So based on this pattern, we divide documents in different topics. This type of modeling is very much useful when there are thousands of documents, and we are trying to get the information based on specific needs. This will take unusual amount time if we try to achieve manually. But topic modeling can satisfy the need in very little time.

Topic modeling is done using Latent Dirichlet Allocation (LDA), one of the popular modeling techniques, and refers to identifying topics that best describes the documents. These topics only will be needed only once required; hence it is called latent [16].

Basics of topic modeling are:

1. Topics will be distributed across all documents
2. Topics will be assigned to a word based on two conditions:
3. What topics are present in the documents.
4. How many times a word has been assigned to a particular topic.
5. We have to repeat the process for all documents.

**Fig. 1 :** Code implementation for LDA

**Pseudo code for LDA [13]:**

For all topics in [1,k] do

Sample mixture components k ~ Dir(Beta)

End for

For all documents m in [1,m] do

Sample mixture proportion m ~ Dir(Alpha)

Sample documents length Nm ~ Poiss()

For all words n in [1, Nm] do

Sample topics index Zm,n ~ Mult(theta)

Sample term for words wm,n ~ Mult()

**Fig 2:** LDA Algorithm [14]

1. Assume there are *k* topics across all the documents
2. Distribute these *k* topics across document *m*(this distribution is known as α and can be symmetric or asymmetric, more on this later) by assigning each word a topic.
3. For each word *w*in document *m*, assume its topic is wrong but every other word is assigned the correct topic.
4. Probabilistically assign word *w*a topic based on two things:  
   - what topics are in document *m*  
   *-*how many timesword *w*has been assigned a particular topic across all of the documents (this distribution is called*β*, more on this later)
5. Repeat this process several times for each document and we are done!

## **Fig 3:** LDA for Topic Modeling

## **The Model [16]:**

Diagram

Description automatically generated

In figure 3, we have what is known as a plate diagram of an LDA model where:

α is the per-document topic distributions,

β is the per-topic word distribution,

θ is the topic distribution for document m,

φ is the word distribution for topic k,

z is the topic for the n-th word in document m, and

w is the specific word

**Fig 4:** Workflow for Topic Modeling

**Diagram

Description automatically generated**

**Text Classification**

A Long Short-Term Memory (LSTM) network is a recurrent neural network differ from traditional feed forward network. LSTM has feedback connection. That’s why, LSTM does not only process single data points but also whole sequence of data. In one LSTM cell, we have input, output, and forget gates. Forget gate tells us how much previous state along with input needs to be considered to update the current state. Basically, we need current input, previous output, and current state to generate new output. In this way, LSTM can process short and long term context.

**Fig 5:** LSTM Diagram at t-time step

Diagram

Description automatically generated

**LSTM Diagram Variable Description**  
Xt = Input vector at the t-time.

Ht−1 = Previous Hidden state.

Ct−1 = Previous Memory state.

Ht= Current Hidden state.

Ct= Current Memori state.

[\*] = multiplication operation.

[+] = addition operation.

So, the input of each LSTM module is Xt(current input), Ht−1, and Ct−1. then the output is Ht, and Ct as seen in figure 5 above.

**Fig 6:** Workflow for Text Classification

**Diagram

Description automatically generated**

**Fig 7:** LSTM Model Workflow

**Timeline

Description automatically generated**

**Sentiment Analysis**

Sentiment Analysis is a technique to predict the feelings and emotions reflected by a word or group of words. Natural Language Processing is the ability of computer to understand human language. For this, we must extract feature from a text to analyses the similarities of the text. Feature extraction process means to extract and produce feature representations that are appropriate for the type of NLP tasks we are trying to accomplish and the type of model we are planning to use. in NLP, feature extraction helps to reduce the amount of redundant data from dataset. In sentimental analysis, we must preprocess the text and make the unstructured data in a form that can be used for classification. In NLP, we need feature extraction techniques to convert the raw text into a matrix or vector of features.

We use a transformer model for sentiment analysis for our project. Transformers architecture is one of the most influential mechanisms in deep learning and NLP. Transformers consist of encoder architecture with positional encoding, inputs, input embeddings, and a block containing neural networks. Transformers architecture is a type of deep learning architecture that learns text representations using a self-attention mechanism. Transformers are the state-of-the-art NLP models that achieve the best performance on many NLP benchmarks. The high-level implementation of transformer models for text classifications are:

a. Importing transformers library

b. Importing the dataset

c. Tokenize the dataset

d. Training the model and making predictions.

e. Evaluate the model

f. Making prediction of test dataset.

For tasks in which the text classes are relatively few, the best performance on text classification can be achieved using pretrained Transformers models like BERT, XLNet, and RoBERT. But transformer models scale quadratically with input sequence length and linearly with the number of classes.

**Fig 8: Transformer Model**

Diagram

Description automatically generated

**BERT Model**

BERT learns contextual relation between words using transformer. Transformer architecture consists of encoder and decoder stack, but BERT uses only encoder part. Encoder learns what is English, what is grammar, more importantly what is context? BERT is stack of encoder and BERT is used to learn language translation, question-answering, sentiment analysis, text summarization etc. All these problems require good understanding of language. Training of BERT is done in two phases. In first phase, BERT model understands the language and context. In second phase, BERT model will solve the problem as it already knew the language in the pretraining phase.

BERT learns language by training two unsupervised tasks simultaneously. One is Masked Language Modeling(MLM) and Next Sentence Prediction(NSP). BERT understands bidirectional context of the sentence in MLM task in pretraining phase. In NSP, BERT takes two sentences and determines whether second sentence follows the first sentence. In this way, BERT understands the context across different sentences themselves. Now, BERT has a very good understanding of language. In fine-tuning, phase BERT delves with specific problem to solve, in our case, that is sentiment analysis.

**Fig 9**: BERT Diagram

Diagram

Description automatically generated

**Fig 10:** Overall Workflow for Sentiment Analysis:

**Diagram

Description automatically generated**

**Dataset**

The dataset is titled “Trip Advisor Hotel Reviews” based on hotels from online reviews of hotels around the world and collected in the TripAdvisor.com database and queried in 2018 as part of research for a paper (Bansal, 2018). The dataset contains two columns “Review” and “Rating” and has 20,491 unique rows of text and ratings scaled 1-5. As seen in Figure 1, we have a visual of the descriptive analysis which provides the ratings have a mean of 3.9 on the scales where we have a minimum of 1 to a maximum of 5 in our ratings data.

**Fig 11 :** Description of ratings and row counts

Table

Description automatically generated

Next, we remove stop words, emojis and special characters and lemmaize the text. run the dataframe column “Review” through the class “Clean” and remove emojis and special characters (Figure 3). Then, the program tokenizes the text and splits the text and counts the number of tokens in the “Review” column.

**Fig 12:** Clean TextGraphical user interface, text, application

Description automatically generated

After, cleaning the text we see that our maximum token length is 2,052 tokens in row 7072, and we have a minimum token count of 7 in row 1501.

**Fig 13 :** Token count

Graphical user interface, text, application, email

Description automatically generated

Next, we perform a word count on the “Top 20 words in Hotel Review” and display it in the chart below. This knowledge will help us gauge review quality a point of reference in terms of validating the applications of this data and assessing the value of the reviews.

**Fig 14: Top 20 Words**

Chart

Description automatically generated

In Figure 5, we get a mean count of 109 tokens for the dataset. We need to make sure that we are gathering enough tokens to derive a business use case for understanding our consumer market and their sentiment towards hotels in general. It is crucial to drive our sentiment analysis, topic modeling, and text classification model with significant token counts per row. In figure 5, it also shows that 50% of the token counts are at or near 80 tokens which is close to our average token count.

**Fig. 15 :** Token length statistics

**Text

Description automatically generated**

**Exploratory Data Analysis**

All data science projects have this one fuel in common: data. For our project, we are using the TripAdvisor Hotel dataset. Every data-based work starts by understanding the data at hand. We have therefore thoroughly “searched” our data to get the more sense of it. To that end, we have massively used visualization as the mean to uncover our data. Suitably chosen charts have made that gives us big to detailed pictures of the data we will be using and will guide us through the next steps of our work.

We gathered the count of all ratings in the dataset. The largest Rating count by far is a Rating of “5” at over 9,000 rows. We have approximately 1,200 ratings of “1”, 1,800 ratings of “2”, and approximately 6,000 ratings of “4”. Generally, the population seems to share positive experiences.

**Fig 16**: Histogram

Chart

Description automatically generated

As a whole we decided to capture the most frequent words through out all of the ratings. These words such as room, crime, police, pool, breakfast, and drink all seem to fit our project features. These features included cost, crime history, amenities, flexibility, parking, indoor facility, room space, air-conditioning, and neighborhood.

**Fig 17:** Word Cloud size based on frequency

Text, application, chat or text message

Description automatically generated

The word frequency within each rating is computed and segregating Ratings from 1 to 5. In the lowest rating in figure 8, we see the top three tokens are police, crime, and stay. In ratings at 2, we see a mixture of police, crime, and good, which could indicate a shift in concerns. Ratings of 3 place good in the top category based on counts, the complimentary positive words have more counts. In the highest rating of 5, crime is nearly at the bottom and surpassed by staff and great. However, we do see that drugs are also in the count. This will require a deeper understanding of the content surrounding the word drug and the sentiment within the review.

**Fig. 18**: Ratings 1 to 5 (starting from top left to bottom )

Chart, bar chart

Description automatically generated

Next, we gathered the most common Bigrams and split them by rating class from 1 to 5.

**Fig** **19** : Exploring bigrams

Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface, table, Excel

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

Table

Description automatically generated

These bigrams and unigrams will be the root of our topic tagging as we decide which bigrams will be most valuable to tagging our reviews. The primary concern here is confirming our analysis is complete by checking the quality and cleanliness of our NLP model. Even after text cleaning there are some tokens that managed to escape the regex filtering and stemming process. We will fine tune our text cleaning, and perform an additional cleaning step involving token removal less than two characters, and decide which NTLK method is best for removing the least valuable tokens.

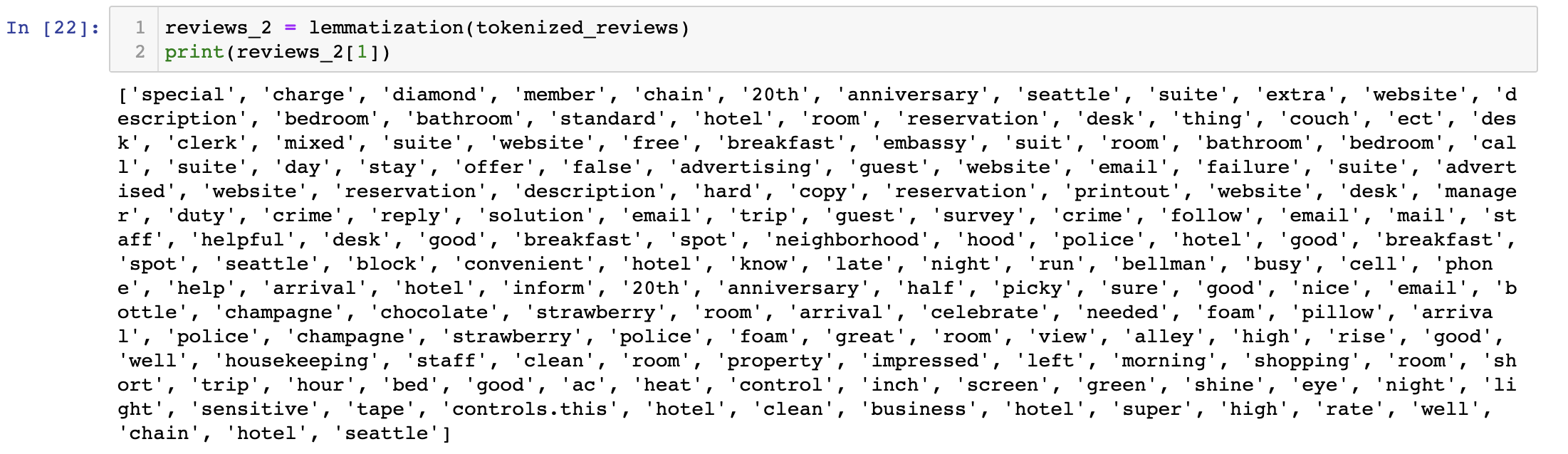
**Feature Extraction**

Successful machine learning model rely on data. Data need to be in good quantity as well as quality. Unfortunately, raw data is almost never appropriate to train a model. In order to make it suitable for model training, data needs to go through a series of steps called globally data processing. In this process, data is cleaned and prepare for model building. One important thing in these steps is the extraction of features. Data are defined by a lot of features or attributes, but not all of them are important to build a specific model. Each model has its requirements and specifications. When it comes to text data as the one we are using, there is a huge amount of words in a language vocabulary. However, not all of them are relevant in the transmission of information. The idea here is to get rid of unnecessary words and keep only those that are meaningful in transmitting information. That is what this section have been focusing on.

**Topic Modeling**

After cleaning and exploring our data we can begin extracting our features to achieve our goal to be build a better model for hotel managers and owners. After loading and cleaning the data, we tokenize the reviews into valuable text for modeling and gather the “full morphological analysis to identify the lemma for each word” (Manning, C., Raghavan, P., & Schuetze, H. (2009).

**Fig 20**: Lemmatize Tokenized Reviews



We created the object for LDA model using genism library and built the LDA model using the doc\_term\_matrix and the library’s dictionary. In figure ( ), we have our 12 topics and the top 10 terms that are associated with each topic.

**Fig 21**: Topic Models

A picture containing table

Description automatically generated

**Text Classification**

In this area, were are starting to lemmatize the reviews and capture text sequences.

**Fig 22:** Lemmatizer

Text

Description automatically generated

We utilize the LSTM from tensorflow keras with recurrent layers in the sequential method for embedding, flattening the layer, and assigning parameters for dropout and density.

After running 20 epochs, we have an accuracy of 75% and loss of 58%. In figure ( ), the validation accuracy and train accuracy are best at epoch 12 where we get 54% and 56% respectively.

**Fig 23**: Text ClassificationAccuracy and Loss

Chart, line chart

Description automatically generated

Next, we run LSTM to capture the mean absolute error and mean squared error loss. The validation line shows that our model performs best at epoch 6.

**Fig 24:** MAE and Loss

Chart, line chart

Description automatically generated

We set aside 20% of the hotel review data for test data, and we have an accuracy of 52%.

**Sentiment Analysis**

We classify reviews and their sentiment into the buckets “neautral”, “positive”, and “negative”. Here are our preliminary results for all 3 classes and their counts in our dataset.

**Fig 25:** Count of Positive, Negative and Neutral reviews

Chart

Description automatically generated

We can see that we have a large portion of positive reviews. Customers may be more enticed to leave positive reviews. First, we must analyze the text to ensure we are getting quality responses. Here are character counts which we can use to help ensure that our review data are generating a balanced insight into consumer experience.

**Fig 26**: Distribution of Characters Across Sentiments

**A picture containing text

Description automatically generated**

**A picture containing table

Description automatically generated**

Graphical user interface

Description automatically generated with low confidence

Next we have the word count per review.

**Fig 27**: Distribution of Words Across Sentiments

Chart

Description automatically generated with medium confidence

Based on these histograms above on “Character Counts” and “Word Counts” per sentiment class, it is safe to assume that reviewers are leaving enough data within all 3 classifications.

In the final portion of sentiment so far we have classified Unigrams, Bi-Grams and Tri-Grams.

**Fig 28:** Most Common Unigrams

Chart, bar chart

Description automatically generated

**Fig. 29:** Most Common Bigrams

Chart, bar chart, histogram

Description automatically generated

A picture containing table

Description automatically generatedChart, bar chart

Description automatically generated

**Fig. 30:** Most Common Trigrams  
Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generatedA picture containing table

Description automatically generated

We used the Bert tokenizer and text sequence classification and captured results on precision to predict the value names for our sample test set. Our model is best at predicting neutral sentiment; however we must recognize that the neutral sentiment was strongly predicted as positive.

**Fig 31:** Confusion Matrix

Graphical user interface, application

Description automatically generated

**Conclusion**

We can conclude after running our data preparation and model predictions that we have a replicable workflow that can be used with new test data and continue to fine tune our parameters. Using the dataset “*TripAdvisor Hotel Review Dataset*” we are make the framework to build three models namely Topic Modeling, Review Rating, and Sentiment Analysis. This could be works as one stop solution for three different workflows in one place. We used Google colab free account to run our notebooks, with a CPU processing speed of 2.2 Ghz and 1 CPU core.   
**Fig 31:** Confusion Matrix

**Fig 31:** Confusion Matrix

**Fig 32**: CPU Specifications  
Text

Description automatically generated

Text

Description automatically generated

**References**

Sentiment Analysis in Hotel Reviews: <https://www.altexsoft.com/blog/sentiment-analysis-hotel-reviews/>

Han, J., Kamber, M. and Pei, J. (2011), Data Mining: Concepts and Techniques, Morgan Kaufmann, Waltham, MA.

Pang, B. and Lee, L. (2008), “Opinion mining and sentiment analysis”, Foundations and Trends in Information Retrieval, Vol. 2 Nos 1-2, pp. 1-135.

Liu, B. (2010), “Sentiment analysis and subjectivity”, in Indurkhya, N., Damerau, F.J. (Eds), Handbook of Natural Language Processing, Vol. 2, Taylor and Francis Group, Boca Raton, FL, pp. 627-666

Khoo, C.S.G., Nourbakhsh, A. and Na, J.C. (2012), “Sentiment analysis of online news text: a case study of appraisal theory”, Online Information Review, Vol. 36 No. 6,   
pp. 858-878

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D. and Kappas, A. (2010), “Sentiment strength detection in short informal text”, Journal of the American Society for Information Science and Technology, Vol. 61 No. 12, pp. 2544-2558.

Wu, X. and Kumar, V. (2008), “Top 10 algorithms in data mining”, Knowledge and Information Systems, Vol. 14 No. 1, pp. 1-37.

Pang, B. and Lee, L. (2005), “Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales”, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 115-124.

Bansal, B. (2018). *TripAdvisor Hotel Review Dataset* [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.1219899>

Manning, C., Raghavan, P., & Schuetze, H. (2009). *Introduction to Information Retrieval*. 581.

[Nguyen, CTH](https://www.researchgate.net/profile/Nguyen-Cao-Truong-Hai). (March 2009). *SVD-LDA: A Combined Model for Text Classification.* Retrieved from <https://www.researchgate.net/figure/A-pseudo-code-for-the-LDA-generation-process_tbl1_220635645>

Doll,Tyler. (July 24, 2018). *LDA Topic Modeling: An Explanation.* Retrieved from

<https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd>

Koçak, Emir. (February, 2022). *In depth series: Sentiment Analysis w Transformers.* Retrieved Nov 28, 2022 at <https://www.kaggle.com/code/emirkocak/in-depth-series-sentiment-analysis-w-transformers>

Pancholi, Nidhay. (June, 2022). *Predicting rating from reviews (Trip Advisor).* Retrieved Nov 28, 2022 at <https://www.kaggle.com/code/nidhaypancholi/predicting-rating-from-reviews-trip-advisor>

Altexsoft. (Mar 12, 2021). *Sentiment Analysis in Hotel Reviews: Developing a Decision-Making Assistant for Travelers.* Retrieved from <https://www.altexsoft.com/blog/sentiment-analysis-hotel-reviews/>

Wikipedia. n.d. *Smoothed LDA* Retrieved from <https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation>