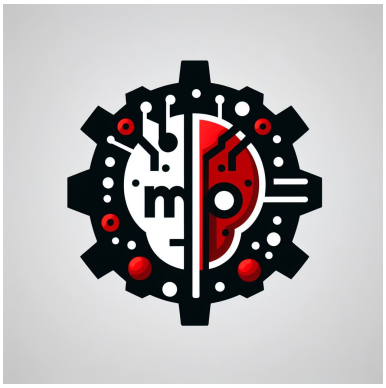
Persian Sentiment Analysis For Marketing



Machine Learning Approach

Department of Mathematics and Compute Science

Team members:

Mobin Nesari

Parmida Jabari

Supervisors:

Hadi Farahani

Shide Sharif

Niloofar Shabani

January 30, 2024

Abstract

Sentiment analysis, also known as opinion mining, is a vital task in natural language processing (NLP) that involves determining the sentiment or emotional tone expressed in textual data. Sentiment analysis has been widely recognized as a critical component of understanding user-generated content on social media platforms and e-commerce websites. With the exponential growth of user-generated content on the internet, sentiment analysis has become increasingly important for businesses and organizations. By analyzing the sentiments expressed in customer reviews, comments, or feedback, companies can gain valuable insights into public opinion, customer satisfaction, and brand perception

The project, starts by analyzing the distribution of happy and sad comments through exploratory data analysis. We then perform text preprocessing and examine common words and factors across different sentiments.Then we aim to classify the sentiments expressed in these reviews into positive and negative categories. We deploy two powerful machine learning models, namely Bidirectional Long Short-Term Memory (BiLSTM) and Convolutional Neural Network (CNN), to unravel intricate patterns and extract sentiments from diverse textual data.

Acknowledgements

We would like to express our sincere gratitude to our supervisors, Professor Hadi Farahani and Teaching Assistants Shide Sharif, Niloofar shabani, for their invaluable guidance and support throughout the development of this movie recommendation system. Their expertise and insights were essential in ensuring the quality and accuracy of the system.

We also extend our thanks to Professor Seyed Ali Katanforush for teaching us data structures and algorithms, which provided the necessary foundation for our understanding of the fundamental concepts that underlie this project.

Finally, we would like to thank Dr. Saeedreza Kherad Pishe for teaching us programming and deep learning, which were critical skills for the development of this recommendation system. Thank you all for your contributions to our education and success.

Contents

Chapter 1

Exploratory Data Analysis

The SnappFood dataset for persian sentiment analysis ,[1] publicly available through the Hooshvare Research, contains user-submitted reviews on SnappFood, a prominent online food delivery service in Iran. The dataset is primarily textual, with reviews ranging in length and complexity. Each entry within the dataset is a discrete piece of feedback provided by a user, reflecting their personal experience with the service.1.1

Table 1.1: Features Data types

Feature	Data Type
comment	object
label	object
label_id	int64

The dataset consists of three main features: 'comment', 'label', and 'label_id'. The 'comment' feature contains the Persian text comments, while the 'label' feature represents the sentiment label of the comment, categorized as 'HAPPY' or 'SAD'. The 'label_id' feature is a numerical representation of the sentiment labels. The dataset is divided into training, testing, and development sets, with the training set containing 56,700 entries, the testing set containing 7,000 entries, and the development set containing 6,300 entries.

In the preprocessing phase, problematic lines in the CSV files were excluded to ensure data integrity. A tab ('\t') delimiter was employed for handling tab-separated values, contributing to a robust and clean dataset for analysis. Notably, the dataset contained no null or duplicate entries, confirming its reliability for analysis.

1.1 Distribution of Happy and Sad Comments

Regarding the distribution of happy and sad comments across the three CSV files, a balanced representation was observed, with a 50-50 distribution. This even distribution contributes to a diverse and representative dataset, allowing for a comprehensive analysis of sentiment within the dataset.1.1

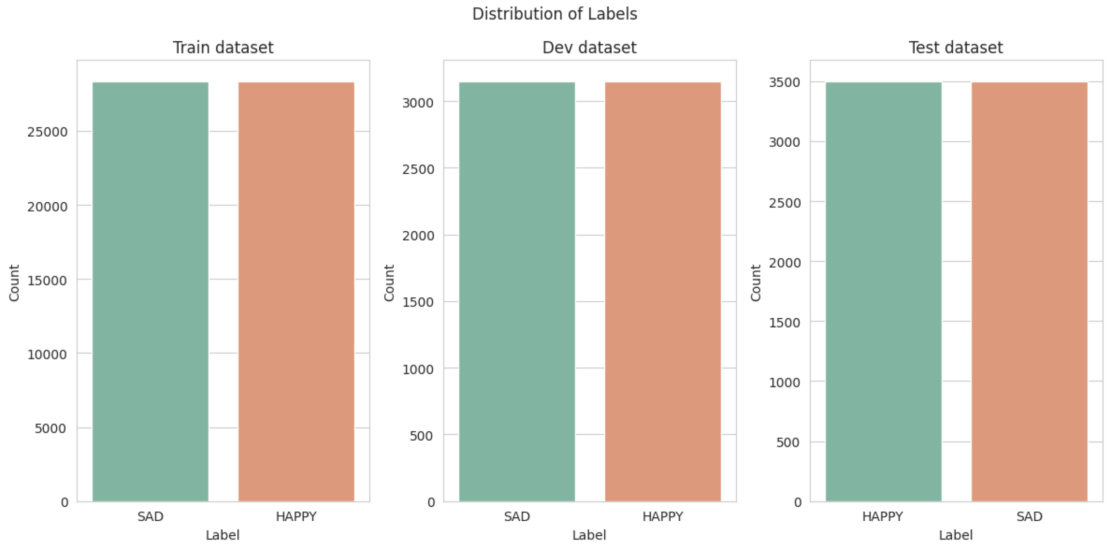


Figure 1.1: Distribution of Happy and Sad Comments

1.2 Text Preprocessing for Persian Sentiment Analysis

In the pursuit of refining the dataset for sentiment analysis, a multi-step text preprocessing approach was implemented. The process began with the removal of non-alphanumeric characters and punctuation from the comment texts using regular expressions. Afterwards, each comment was tokenized into a list of words using the Natural Language Toolkit (NLTK) library.

To further enhance the quality of the dataset, a set of Persian stop words, curated through the Hazm library, was employed. These stop words, indicative of common and non-informative terms, were eliminated from the tokenized comments. This step helps in focusing the analysis on the essential sentiment-bearing words, resulting in a more meaningful representation of the underlying sentiments in the dataset.

The resulting dataset, now enriched with cleaned and refined comments, is ready for more effective sentiment analysis, where the impact of irrelevant noise has been significantly removed. This text preprocessing pipeline sets the stage for a robust and accurate exploration of sentiment trends within the Persian language corpus.

1.3 The Most Common Words in Comments

The word cloud and TreeMap diagrams collectively pinpoint the recurring presence of essential words, prominently featuring most occurred words in the lexical tapestry of our dataset. These words, although seemingly commonplace, bear significant weight in shaping the overall narrative and emotional undertones within the comments.1.2

Figure 1.2: The Most Common Words in Comments

1.4 Analysis Of Different Factors Across Sentiments

In this section, we examine the distribution of comment lengths in the dataset, discerning patterns that distinguish happy and sad sentiments. A visual exploration is conducted to highlight the disparities in comment lengths between happy and sad sentiments. The goal is to uncover potential insights into the linguistic characteristics associated with each sentiment category.1.3

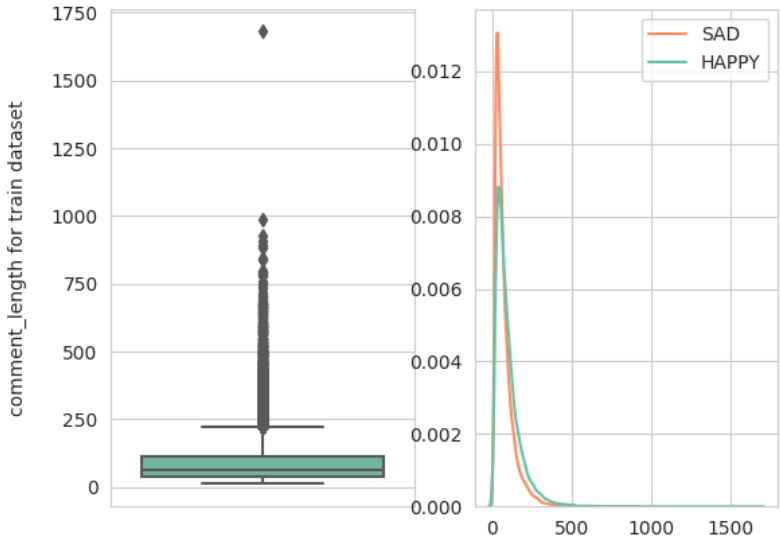


Figure 1.3: Representation of

This analysis delves into word counts within comments, offering insights into the expressive diversity associated with happy and sad sentiments. Variances in word counts suggest nuanced linguistic patterns, holding significance for sentiment analysis and the enhancement of Persian sentiment prediction models. 1.4

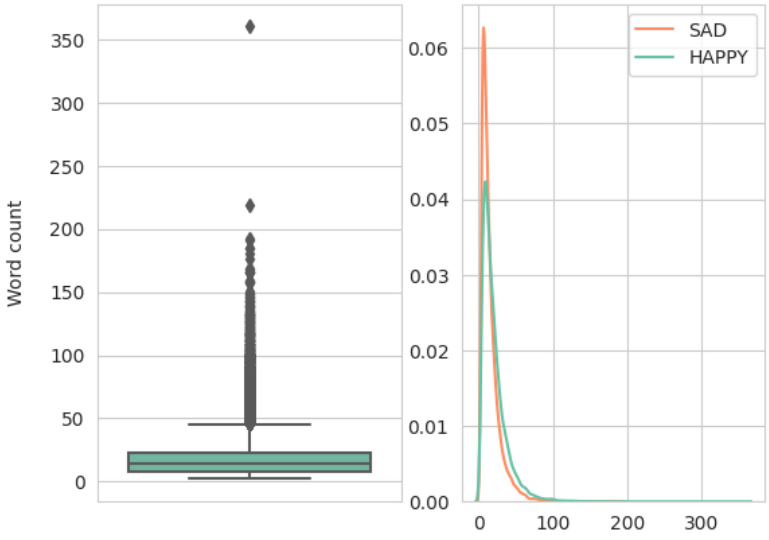


Figure 1.4

Chapter 2

Methodology

After a comprehensive Exploratory Data Analysis (1), the focus shifted to preparing the Snappfood dataset for in-depth analysis. This crucial step involved handling of the textual data to ensure that it's ready for deeper analysis. The key here is to make sure that our sentiment analysis model is based on good, reliable data.

2.1 Enhancing Sentiment Analysis through Rigorous Text Preprocessing

We started by shuffling the dataset, introducing a randomized order to the records. This ensures a balanced representation in both training and testing sets, minimizing biases that might arise from specific data arrangements. Additionally, a binary labeling scheme was applied to facilitate streamlined binary classification, a crucial simplification for subsequent modeling.

Textual data often exhibits irregularities and variations that can hinder effective analysis. To address this, a comprehensive normalization process was implemented using the Hazm Normalizer, contributing to the standardization of the text. Tokenization further disassembled the text into individual units, allowing for a more granular exploration of semantic nuances.

The integration of pre-trained FastText word embeddings brought a transformative dimension to our study. These embeddings encapsulate rich semantic information about individual words, harnessing contextual relationships within the language. By aligning our dataset with these embeddings, we elevated the representation of words in our model, enhancing its capacity to understand and generalize

from the input data.

A pivotal aspect of our approach was the creation of an embedding matrix that encapsulates the semantic essence of our vocabulary. This matrix, with dimensions mirroring the vocabulary size and FastText embedding size, serves as a bridge between our dataset and the pre-trained embeddings. Incorporating this matrix into our model ensures that the learning process capitalizes on both the intrinsic characteristics of our data and the wealth of information embedded in the pre-trained vectors.

The tokenization process, facilitated by the Keras Tokenizer, converted our textual data into sequences of integers, a prerequisite for neural network training. The introduction of padding ensured a consistent input length, addressing the variability in sentence lengths and creating a structured foundation for our model.

2.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs), play a crucial role in the field of artificial intelligence, introducing a dynamic element that enhances the processing of sequential data. Unlike traditional neural networks, RNNs possess a memory mechanism, allowing them to retain information from previous steps and consider it alongside current inputs.

The versatility of RNNs extends to a wide array of applications. From predicting stock prices to understanding the nuances of language in text, these networks work best in scenarios where context and temporal dependencies matter. The ability to retain information from the past allows RNNs to tackle tasks that involve not just individual data points, but the context and order in which they unfold. The RNN formula can be expressed as:

$$h_t = f_W(h_{t-1}, x_t) = tanh(W_{hh}h_{t-1} + W_{xh}X_t)$$

where h_t is updated hidden state at a specific time step. It combines information from the previous hidden state(h_{t-1}) and the current input(X_t) using tanh activation to capture sequential dependencies.

2.3 Transformers

Transformer models are a type of deep learning model that is used for natural language processing (NLP) tasks. Introduced in a paper in 2017, transformers have

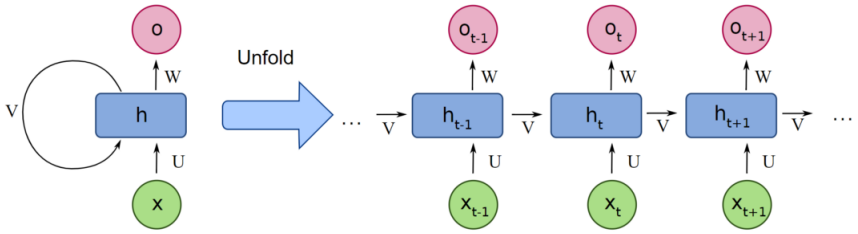


Figure 2.1: General RNN Architecture

become the state-of-the-art for various NLP tasks, including language translation, text summarization, question answering, and more.

The key idea behind transformers is the concept of attention, which allows the model to focus on different parts of the input sequence while generating the corresponding output sequence.

Transformer models consist of a dual-phase process. Initially, the input sentence encodes into a sequence of vectors. This encoding process uses a self-attention mechanism, a key component enabling the model to understand the relationships among the words in the sentence. Following the encoding phase, the model proceeds to decode the input sentence into a sequence of output tokens, once again relying on the self-attention mechanism.

In our study on sentiment analysis using transformers for the Snappfood dataset, helped us to understand complex language structures and subtle context hints in user comments. 2.2

2.4 Our Sentiment Analysis Models

For our sentiment analysis on Snappfood comments, we used two advanced deep learning models: Bidirectional Long Short-Term Memory (BiLSTM) and Convolutional Neural Network (CNN). These models are carefully crafted to uncover patterns and understand sentiments in the diverse textual data.

Our BiLSTM model tends to understand the sentiment in Snappfood comments by capturing sequential patterns effectively. Starting with an embedding layer using pre-trained FastText word embeddings, it enriches the input with semantic information.The core of the BiLSTM model is the Bidirectional LSTM layer with 300 units. This layer looks at the context both forward and backward, helping the model comprehend the comment's overall meaning.

Global Max Pooling is used to focus on essential features. This layer picks out the

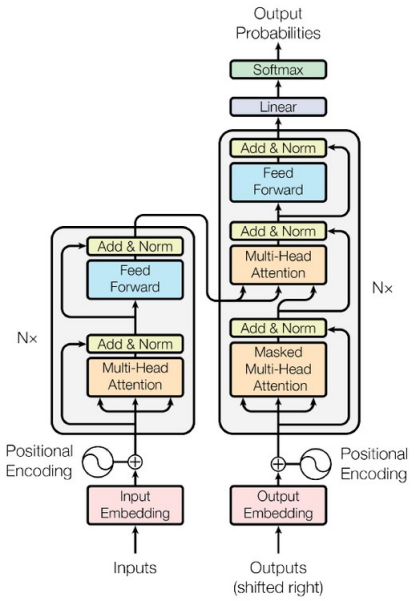


Figure 2.2: Transformers Architecture

most important features from the sequential output, helping the model concentrate on main aspects of the input, leading to better sentiment analysis. Dense layers with dropout prevent overfitting, helping the model generalize. The final output is binary, indicating sentiment probability for each comment.

Our CNN model is designed to uncover different layers of meaning in Snappfood comments, enhancing sentiment analysis. The model starts with an embedding layer using fixed FastText embeddings. These embeddings is the basis for the model to find fundamental word meanings.

Declared CNN design consists of three layers with different filter sizes (4, 8, and 16), allowing the model to explore different textual features, from local to broader patterns. After each convolutional layer, a ReLU activation function leads to non-linearity.

To focus on essential features, we employ max pooling and global max pooling layers for dimensionality reduction. The following dense layers, complemented by dropout for regularization, lead to the binary classification output. The sigmoid activation function outputs the sentiment probability score for each comment.

Both the BiLSTM and CNN models are trained over 10 epochs, a carefully chosen number of iterations to balance training efficiency and model convergence. The training process involves minimizing the binary cross-entropy loss, optimizing the model's ability to classify comments accurately. Evaluation on the testing set shows

the models' performance on unseen data, providing insights into their generalization capabilities.

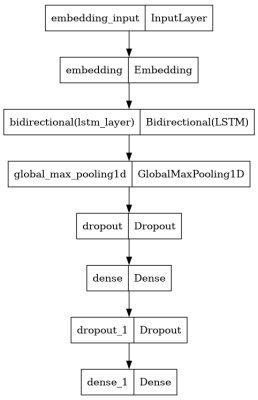


Figure 2.3: B-LSTM Model Summary

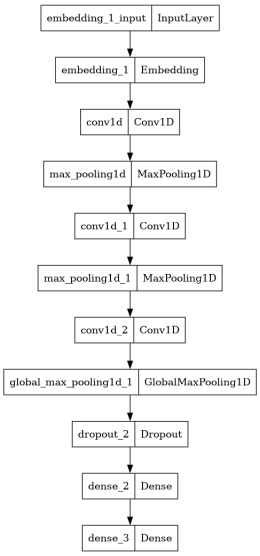


Figure 2.4: CNN Model Summary

Chapter 3

Conclusion

3.1 Results

The results of the persian sentiment analysis shows a good understanding of the sentiments expressed in user comments. Both of the implemented Bidirectional Long short term memory (BiLSTM) and Convolutional Neural Network (CNN) models perform well on discerning between positive and negative sentiments. In the training phase, the models capture sequential patterns and diverse textual features effectively. The performance evaluation on the testing set shows the models' capability to generalize well to unseen data. The achieved accuracy and other relevant metrics showcase the effectiveness of the models in classifying sentiments accurately. The Bidirectional Long Short-Term Memory (BiLSTM) model achieves a test accuracy of 83.99%, while the Convolutional Neural Network (CNN) model follows with an accuracy of 82.69%. Evaluating the models using the weighted F1 score shows the performance of the BiLSTM model, with a score of 0.84. The weighted F1 score of 0.83 in the CNN model also shows good performance of this model. Analyzing the confusion matrices allows us to assess the models' strengths and weaknesses in differentiating positive and negative sentiments. 3.23.1

3.2 Future Works

There are several task which may declared on this dataset. As you know, Snapp Food is a big food delivery platform and for a company like it, chat-bots play a crucial part in their business. A good chat-bot should recognise customer's intent from discussion so a great task which can be implemented is intent classification.

Another future work that may be pursued is using this model on other datasets

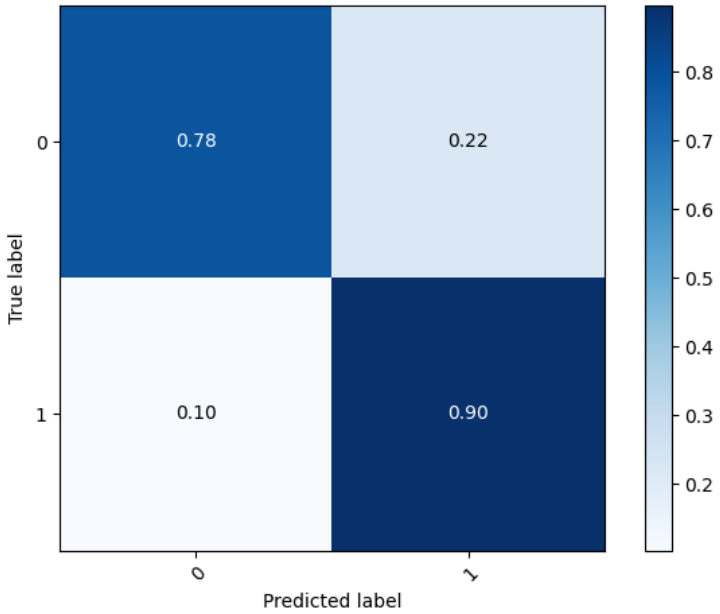


Figure 3.1: B-LSTM Confusion Matrix

which has comments like Digikala, Tapsi or even application markets like Sibche. As we talked in introduction, where we have comments like movie reviews or product reviews, sentiment analysis plays a critical role in that section.

3.3 Accessing the Code

To access the code for this project, please visit the following GitHub repository: https://github.com/MobinNesari81/SnappFood-Sentiment-Analysis.

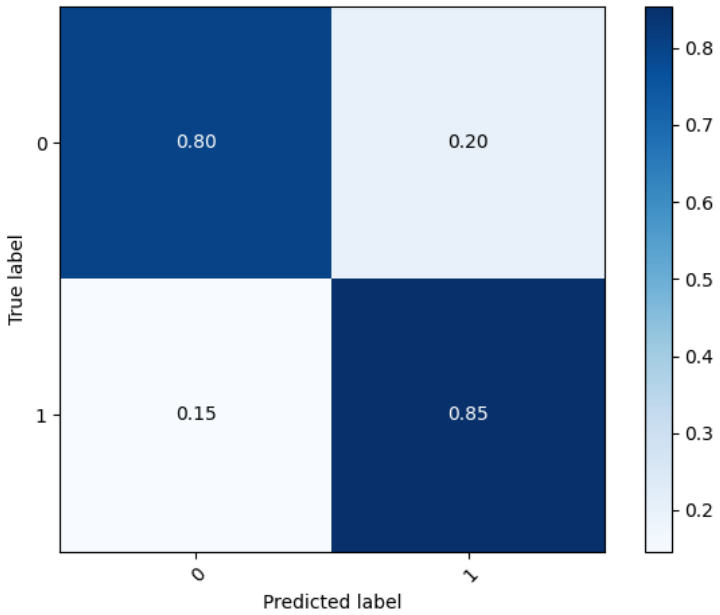


Figure 3.2: CNN Confusion Matrix

Bibliography

[1] Kaggle. (n.d.). *SnappFood Dataset*. https://hooshvare.github.io/docs/datasets/sa#snappfood