Pattern Recognition (CEN474)

Project - Final Report

Project title: Sentiment Analysis

Abstract: This project implements a sentiment analysis system that classifies user reviews as positive or negative. By utilizing deep learning models with an embedding layer and Gated Recurrent Units (GRUs), the system processes text data to predict sentiment. The application is interactive, employing a user-friendly interface built with Tkinter, enabling real-time sentiment predictions. This report provides an overview of the problem, methodology, and findings of the project.

Keywords: Sentiment Analysis, Natural Language Processing, Deep Learning, GRU, Embedding, Text Classification, Tkinter, Keras, Tensorflow, Numpy, Pandas.

1. **Introduction:** Sentiment analysis is a key application of Natural Language Processing (NLP) that interprets and classifies human emotions in text. In the context of e-commerce, understanding customer reviews can help businesses improve services and products. This project aims to develop a sentiment analysis system for classifying customer reviews as positive or negative, using deep learning models for accurate predictions.

2. **Literature Reviews/ Related Works:**

- **Analysis of Amazon Reviews with GRU:** One study classified Amazon shoe reviews as positive or negative using GRU and LSTM-based models [29]**.**
- **Classification of IMDB Comments Using GRU:** A study of the IMDB dataset revealed that the GRU and bidirectional GRU models successfully capture sequential dependencies in review data [31] [33].
- **Analysis of Social Media Posts with GRU:** A study on social media data showed that GRU-based models can effectively analyze the sentiment and meaning levels of texts [32].

3. **Problem Statement:** Given the increasing volume of user-generated content, it is crucial to develop efficient models to analyze sentiments in textual data. The challenge lies in accurately predicting sentiment polarity in reviews while handling the variability and noise inherent in natural language. This project addresses these challenges using a GRU-based model.

4. **Contributions:**

    - Developed a GRU-based sentiment analysis model optimized for the Hepsiburada dataset.
    - Implemented an interactive graphical user interface for real-time predictions.
    - Provided insights into model performance and limitations for future improvements.
    - Implemented preprocessing techniques, including tokenization and padding, to handle text variability.
    - Evaluated the model on real-world data, demonstrating its applicability in e-commerce.

5. **System Model or Proposed Methodology:**

- **Data Preprocessing:** Reviews were tokenized using Keras's Tokenizer with a vocabulary size of 10,000 words. The sequences were then padded to a fixed length determined by the mean and standard deviation of token lengths.

- **Model Architecture:**
    - An Embedding layer maps tokens to a 50-dimensional vector space.
    - Four stacked GRU layers with decreasing units (32, 16, 8, 4) model the temporal dependencies in the data.
    - A Dense output layer with a sigmoid activation function performs binary classification.
    - It has 1 neuron to produce a value, if the value is near 0, it means the review is negative, if the value is near 1, it means the review is positive.
    - Because the RNN's has problems of Exploding and vanishing gradient, we have used LSTM and GRU.

- **Training:** The model was trained using customer review data with an 80-20 train-test split. Our **Accuracy** is 95.12% in this model.
- **Evaluation:** Model performance was assessed using accuracy metrics.
- **Interface Development:** A Tkinter-based GUI for end-user interaction.

6. **Mathematical system Model:**
   We have used:
    - Loss Functions
    - Activation Functions
    - Optimization Algorithms

- Tokenization and Padding steps
- Keras's Embedding Layer
- GRU's mathematical formulation

- **Tokenization and Padding**

  Tokenization: Converts words in the text into indices:

  x=[x1,x2,...,xn] (Tokenized word sequence) Here, xi is the index of a word.

  Padding: Ensures that all input sequences have the same length by padding with zeros

  Xpad=[x1,x2,…,Xmax]

  Here, sequences are padded to the length max_tokens

- **Word Embeddings (Embedding Layer)**

  The code uses Keras's Embedding Layer to represent word indices in a dense vector space:

  E : W → $\boldsymbol{R^d}$

  W: Number of words

  d: Embedding vector size (embedding_size = 50).

- **GRU Layers**

  GRU's mathematical formulation:

  z_t=σ($W_z x_t$+$U_z h_{t-1}$+b_z)(Update gate)

  r_t=σ($W_r x_t$+$U_r h_{t-1}$+b_r)(Reset gate)

  $\hbar_t$=tanh($W_h x_t$+$U_r h_{t-1}$+b_r)(Candidate state)

  h_t=$z_t \odot h_{t-1}$+(1-z_t)$\odot\ \hbar_t$ (Final hidden state)

  Where:

  - x_t: Input at time step t,
  - h_t: Hidden state,
  - z_t, r_t: Gates,

- σ :Sigmoid activation, tanh: Hyperbolic tangent activation.

- **Output Layer(Danse Layer)**

  The final layer uses the **sigmoid activation function** to output a probability value between 0 and 1:

  ¥ = σ($w^t h + b$) = $\frac{1}{1+e^{-(w^t h+b)}}$

  If ¥ > 0.5, the statement is classified as positive, otherwise negative.

- **Loss function and optimization**

  Loss Function: Binary classification uses **binary cross-entropy** loss:

  L = -$\frac{1}{N}\sum_{i=1}^{n}[y_i \log(t_i) + (1 - y_i)\log(1 - t_i)]$

  Where:

  N: appx 200.000 (Training Data)

  - y_i: True label (0 or 1),
  - t_i: Predicted probability.

  Optimization: Weights are updated using the Adam algorithm:

  w:=w−η∇w_L

  η: Learning rate.

  ∇w_L: $Loss\ Gradient$

7. **Problem Solution:** To solve the problem, the model was trained using labeled review data. The GRU model was trained for 8 epochs with a batch size of 128. Data preprocessing ensured uniform input dimensions, and the Adam optimizer accelerated convergence. Validation accuracy confirmed the model's ability to generalize to unseen data.

8. **Results and Discussion:** The model achieved a test accuracy of approximately 95%. The evaluation highlighted its capability to capture the nuances of sentiment in reviews. The

most common errors involved ambiguous reviews or those with conflicting sentiments. Visualization of GRU activations revealed the model's focus on critical phrases.

9. **Comparison of your work with related works**

- Unlike traditional models such as Naïve Bayes and SVM, which rely on handcrafted features, the GRU-based approach automatically learns and captures sequential patterns in text data. This aligns with the findings of Chen et al., who highlighted GRU's capability to outperform simpler RNN architectures in capturing contextual information [30].
- A study on Amazon reviews emphasized the effectiveness of GRU in handling noisy and lengthy textual data, a challenge that our model also addressed with padding and tokenization techniques [29].
- GRU models have also proven effective in domain-specific tasks like IMDB sentiment analysis. Similarly, our model leverages GRU's strength in sequential data processing to handle diverse review content [31] [33].

Overall, the proposed model strikes a balance between computational efficiency and predictive performance, making it well-suited for e-commerce applications.

10. **Conclusion:** This project successfully implemented a GRU-based sentiment analysis model for e-commerce reviews. The results validate the efficacy of GRU networks in text classification tasks.

11. **Limitations of your work:**

- The model's performance is limited by the size and quality of the dataset.
- Ambiguity in reviews affects prediction accuracy.
- The model does not account for multi-class sentiment.

12. **Future direction of your work**

- Expanding the dataset to include reviews from diverse sources.
- Exploring multi-class sentiment analysis.
- Implementing attention mechanisms to improve interpretability.

References:

[29] https://github.com/Rittikasur/Sentimental-Analysis-using-LSTM-GRU

[31] https://github.com/meghanshgarjala/SENTIMENT-ANALYSIS-USING-LSTM-GRU

[33] https://ieeexplore.ieee.org/document/10174396

[32] https://www.ecs.csun.edu/

https://en.wikipedia.org/wiki/Sentiment_analysis#:~:text=Sentiment%20analysis%20(also%20known%20as,affective%20states%20and%20subjective%20information.

https://www.geeksforgeeks.org/what-is-sentiment-analysis/

https://www.analyticsvidhya.com/blog/2022/07/sentiment-analysis-using-python/