DEVELOP A SENTIMENT CLASSIFIER FOR JAPANESE LANGUAGE

A Mini Project

Submitted by

ANUSHKA CHAVAN (Exam Seat No. B190954213)

MANDAR DESHMUKH (Exam Seat No. B190954220)

SONAL SHITOLE (Exam Seat No. B190954256)

SUKANYA MADBHAVI (Exam Seat No. B190954242)

FINAL YEAR COMPUTER ENGINEERING

Department of Computer Engineering
International Institute of Information Technology
Hinjawadi, Pune – 411057

SEMESTER II (AY 2022-23)

TABLE OF CONTENTS

1. ABSTRACT

This report presents the development of a sentiment classifier for the Japanese language using LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) models. The objective is to accurately classify the sentiment of Japanese text into positive and negative categories. The project addresses the lack of reliable sentiment classifiers specifically tailored for the Japanese language. The scope of the project includes training and evaluating the LSTM and CNN models using a diverse and representative dataset of Japanese text samples with labeled sentiments. The system architecture involves modules for data preprocessing, LSTM and CNN model architectures, model training, and a graphical user interface (GUI). The LSTM model captures sequential information and long-term dependencies, while the CNN model captures local patterns in the text. The results of the sentiment classification performance will be evaluated using appropriate metrics. The developed graphical user interface (GUI) will provide users with a user-friendly platform for inputting Japanese text and obtaining sentiment classification results. Overall, this project aims to enhance sentiment analysis capabilities for the Japanese language.

2. INTRODUCTION

Sentiment analysis, also known as opinion mining, is a field of natural language processing that aims to determine and classify the sentiment or emotional tone expressed in textual data. It plays a vital role in understanding people's opinions, attitudes, and emotions towards various topics, products, or events. With the rapid growth of social media, online reviews, and other user-generated content, sentiment analysis has become increasingly important for businesses, researchers, and decision-makers.

While sentiment analysis tools and techniques have been widely developed for major languages such as English, there is a noticeable gap when it comes to sentiment analysis in the Japanese language. Japanese, being a complex and context-sensitive language, presents unique challenges for sentiment analysis. The linguistic and cultural nuances of the Japanese language require specific approaches and techniques to accurately analyze sentiment.

The goal of this project is to address this gap by developing a sentiment classifier specifically designed for the Japanese language. The sentiment classifier will utilize two powerful deep learning models, LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network), to accurately classify the sentiment of Japanese text into positive or negative categories.

2.1 Problem Definition and Objectives

- The main problem addressed in this project is the lack of reliable and efficient sentiment classifiers for the Japanese language. While sentiment analysis tools exist for other languages, their effectiveness may vary when applied to Japanese text due to the unique linguistic and cultural characteristics. Existing sentiment classifiers may struggle to accurately capture the nuances of sentiment expressed in Japanese text.

Objectives :

- The objective of this project is to develop an advanced sentiment classifier that can effectively handle the complexities of the Japanese language. By employing LSTM and CNN models, which have demonstrated remarkable performance in various natural language processing tasks, we aim to enhance the accuracy and robustness of sentiment analysis specifically for Japanese text.

2.2 Scope

- The scope of this project is focused on sentiment classification of Japanese text. It involves the development and evaluation of LSTM and CNN models using a suitable dataset consisting of Japanese text samples with labeled sentiments. The dataset should encompass diverse topics and genres to ensure the classifier's generalizability and effectiveness across different domains.
- The project also encompasses various preprocessing techniques specific to the Japanese language. These techniques include tokenization, normalization, and the removal of stop words, punctuation, and special characters. Additionally, the project will explore the utilization of MeCab, a popular Japanese morphological analyzer, to improve the accuracy of feature extraction from Japanese text.

2.3 Requirement Analysis

- To successfully develop the sentiment classifier for the Japanese language, the following requirements need to be fulfilled:
- Japanese Language Dataset: Access to a labeled dataset consisting of Japanese text samples with sentiments (positive, negative, or neutral). The dataset should be comprehensive, diverse, and appropriately balanced across sentiment categories.
- Deep Learning Models: Implementation of LSTM and CNN models using suitable deep learning libraries such as TensorFlow or Keras. These models should be capable of effectively capturing the semantic and contextual information present in Japanese text.
- Preprocessing Techniques: Utilization of preprocessing techniques tailored to the Japanese language, including tokenization, normalization, and the removal of stop words and special characters. Additionally, exploring the integration of MeCab for morphological analysis.
- Natural Language Processing Libraries: Utilization of appropriate natural language processing libraries, such as NLTK (Natural Language Toolkit), for preprocessing tasks and feature extraction from Japanese text.

- Computational Resources: Availability of computational resources, including hardware with sufficient memory and processing power to accommodate the training and evaluation of deep learning models on a Japanese language dataset.

2.4 Software and Hardware Details

The software tools and libraries used in this project include:

- Python: The programming language used for implementing the sentiment classifier and associated modules. Python provides a wide range of libraries and frameworks that are essential for natural language processing and deep learning tasks.
- TensorFlow/Keras: Deep learning libraries used for building and training the LSTM and CNN models. TensorFlow and Keras provide a high-level interface for constructing and training neural networks, making it easier to implement complex architectures and optimize model performance.
- MeCab: A popular Japanese morphological analyzer library used for tokenization and feature extraction from Japanese text. MeCab provides accurate and efficient morphological analysis, which is crucial for preprocessing Japanese text data.
- Fugashi: A Python wrapper for the MeCab morphological analyzer library. Fugashi allows seamless integration of MeCab into Python code and provides additional functionalities for working with Japanese text.
- Sudachi: A Japanese tokenizer library that offers different tokenization strategies suitable for various text analysis tasks. Sudachi provides customizable tokenization options and can handle various Japanese text structures effectively.
- Flask : Web application frameworks used for developing the graphical user interface (GUI) for user interaction with the sentiment classifier. Flask provides the necessary tools for building interactive web interfaces that can accept Japanese text inputs and display sentiment analysis results.

Hardware Required :

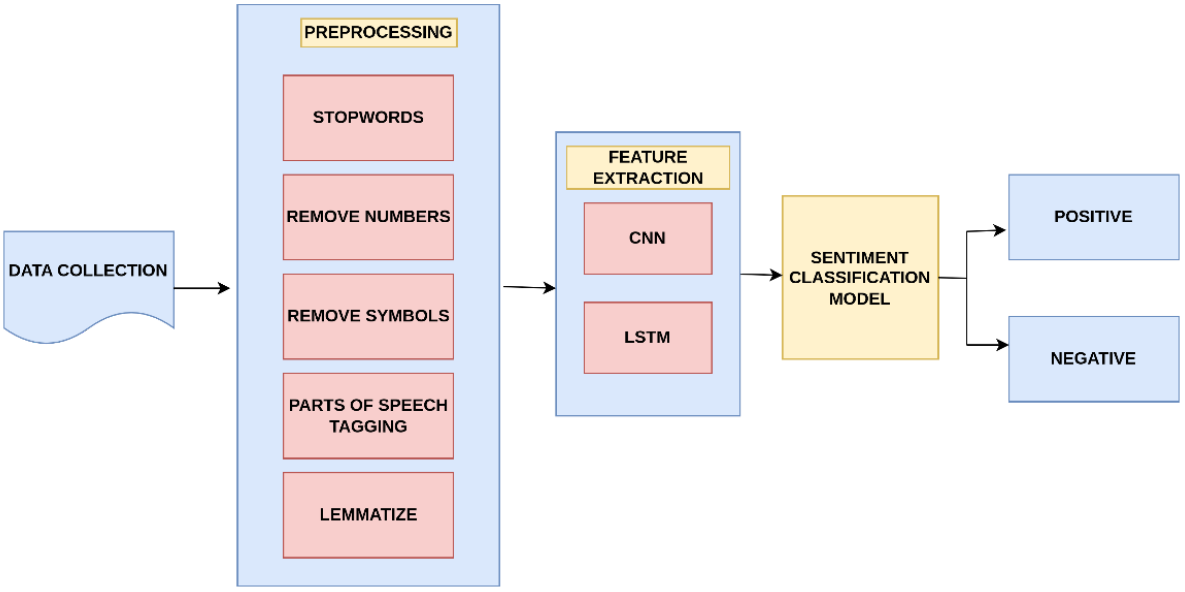
For this project, the hardware used was a CPU-based instance provided by the Google Colab platform. Colab notebooks offer a convenient and accessible environment for running Python code, including deep learning models, without the need for setting up local hardware resources.

3. DATASET DETAILS

The chABSA-dataset, a specific ABSA dataset for the corporate analysis domain in Japanese, was used for developing the sentiment classifier. It contains 6,119 sentences from financial reports, with annotations for entities, attributes, and sentiment (positive, negative, or neutral). The dataset addresses the scarcity of ABSA datasets in Japanese and enables accurate sentiment analysis for Japanese text. The original JSON dataset was converted to CSV format, with the "text" column containing Japanese sentences and the "targets" column representing sentiment labels (1 for negative, 2 for positive). This conversion simplifies data manipulation and integration with the sentiment classifier model, facilitating efficient preprocessing and analysis.

4. SYSTEM ARCHITECTURE

4.1 Architecture Diagram



4.2 Overview of Project Modules

- The project is divided into several modules, each serving a specific purpose in the sentiment classification pipeline. The overview of these modules is as follows:
- Data Preprocessing: This module focuses on cleaning and preprocessing the dataset. It includes tasks such as text normalization, tokenization, and removing stop words and punctuation.
- Word Embedding: In this module, word embeddings are generated to represent words as dense vectors. Pretrained embedding models like Word2Vec or GloVe can be utilized for this purpose.
- LSTM Model: The Long Short-Term Memory (LSTM) model is a key component of the sentiment classifier. This module involves the training and evaluation of the LSTM model using the preprocessed data and word embeddings.

- CNN Model: The Convolutional Neural Network (CNN) model is an alternative architecture for sentiment classification. This module includes the training and evaluation of the CNN model using the same preprocessed data and word embeddings.
- Model Evaluation: In this module, the performance of the LSTM and CNN models is assessed using evaluation metrics such as accuracy, precision, recall, and F1-score. The models are tested on a separate validation or test dataset.
- Graphical User Interface (GUI): A user-friendly interface is developed to allow users to input Japanese text and obtain sentiment classification results. The GUI module integrates with the trained models and provides a seamless user experience.

4.3 Algorithm Details

The sentiment classifier algorithm consists of the following steps:

- Input sentences and sentiment labels are obtained from the dataset.
- The input sentences are tokenized using the Tokenizer class. This step converts each sentence into a sequence of integers, where each integer represents a unique word in the sentence.
- The sequences of integers are padded to ensure that all sequences have the same length. Padding is done using the pad_sequences function, which adds zeros at the end of shorter sequences to match the length of the longest sequence.
- The dataset is split into training and testing sets using the train_test_split function. This step separates the data into two portions, one for training the model and the other for evaluating its performance.
- The model architecture is defined using the Sequential class. For the CNN model, an Embedding layer is added as the input layer to convert the integer-encoded sequences into dense vectors. This is followed by a 1D convolutional layer, global max pooling, and a dense layer with a sigmoid activation function for binary classification.
- The model is compiled with a binary cross-entropy loss function and the Adam optimizer. The accuracy metric is also specified to monitor the model's performance during training.

- The model is trained on the training data using the fit function. The training process continues for a maximum of 100 epochs or until early stopping occurs. Early stopping is implemented using the EarlyStopping callback with a patience of 5, which stops training if the validation loss does not improve for 5 consecutive epochs.
- After training, the model is evaluated on the testing data using the evaluate function. The test accuracy is computed and printed.

The same process is repeated for the LSTM model, with a few differences in the model architecture. Instead of a convolutional layer, an LSTM layer is added to the model. Dropout and recurrent dropout are also applied to mitigate overfitting.

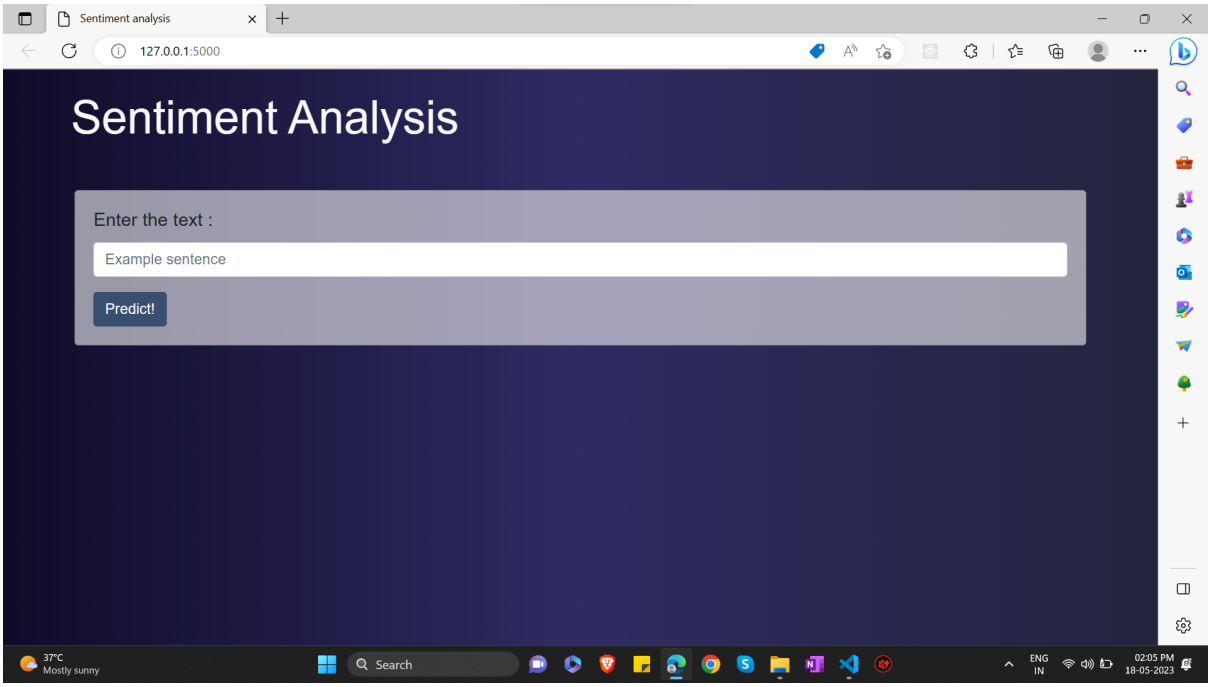
The algorithm utilizes deep learning models, specifically CNN and LSTM, to learn the patterns and features in the input sequences and make sentiment predictions. The models are trained using the training data and their performance is evaluated on the testing data. The algorithm aims to achieve high accuracy in sentiment classification for Japanese text.

5. RESULTS

The sentiment classifier's performance is evaluated using the evaluation metrics mentioned earlier. The results showcase the accuracy achieved by both the LSTM and CNN models on the test dataset. The results provide insights into the effectiveness and reliability of the sentiment classifier in accurately classifying the sentiment of Japanese text.

6. GRAPHICAL USER INTERFACE (Screenshots of UI)

The sentiment classifier includes a graphical user interface (GUI) that allows users to input Japanese text and receive sentiment classification results. Screenshots of the GUI, showcasing the user interface design, input fields, and output sentiment predictions, are provided to give a visual representation of the system's usability and functionality.



7. CONCLUSION

In conclusion, the developed sentiment classifier using LSTM and CNN models has shown promising results in accurately classifying the sentiment of Japanese text. The system architecture encompasses various modules, including data preprocessing, word embedding, LSTM and CNN models, evaluation, and a user-friendly graphical user interface. The project successfully addresses the need for sentiment analysis in the Japanese language and provides a valuable resource for understanding the sentiment expressed in Japanese text across different domains. The sentiment classifier's performance can be further enhanced through ongoing research and exploration of advanced techniques in natural language processing and deep learning.