CLASSIFICATION OF POETRY TEXT INTO EMOTIONAL STATES

1D.Sunil, 2A. Sri Sushma, 3B.Saiteja, 4A.Sharmila, 5G.Sri Ramya

1Assistant Professor, Department Of CSE, Lendi Institute of Engineering & Technology,
2,3,4,5Student – 8th Semester Department of Computer Science & Engineering
Lendi Institute of Engineering and Technology, Jonnada, Vizianagaram

Abstract: Poetry is widely regarded as a genre in which semantic and formal components of language are interrelated in a particular meaningful way. Emotion classification from poetry or formal texts has received less attention by the experts of computational intelligence in recent times. Most of the work has been carried out on classifying emotions from informal text such as chat, sms, email and online user reviews. Previously it used data mining algorithms. In this , an emotion state classification system for the text is using the latest technology of artificial intelligence , called deep learning. For this we are using an attention based BI-LSTM (Bidirectional-long short term memory) model which is a sequence processing model that consists of two LSTM’s, along with a gated recurrent unit which uses gates to control the flow of information that will be implemented on the text corpus. The main aim of the objective is to classify the text into different emotional states like joy, fear, sadness and anger.

Key Words: Deep Learning, Emotion Recognition, Text, Attention-based Bi-LSTM

I. INTRODUCTION:

The classification of opinions, sentiments and emotional states has gained the attention of experts from different fields like natural language processing, computational linguistics and computational intelligence. There are two types of writings that can be analyzed by machine: formal and informal. The formal textual content pertains to poetry, novels, essays, novel, plays, and official/legal documentation, whereas the informal textual content is about SMS, chat, and social media posts.

Poetry is when an emotion has found its thought and the thought has found words. Even for a poet as intellectual as Robert Frost, poetry, at its base, is emotion. He also said, ‘Poetry begins in delight and ends in wisdom.’ In other words, emotion is the basis of poetry and the deep hidden insights that your teachers desire from you - those are the end products. In this lesson, rather than jumping to the profound meanings in poetry, I’m going to show you how to get in touch with the emotions and thus get to the origins of poems

Due to the complex nature of the formal text, detection and classification of emotional states is a challenging task. For instance, the verse and the sunlight claps the earth, and the moonbeams kiss the sea’”, taken from the poem “Love Philosophy” (Shelley) conveys a love emotion. The manual strategy for detecting emotional states expressed by the poet in the poetry text is difficult and time-consuming.

In recent times, machine learning techniques have been applied successfully for extracting and analyzing emotional states and themes from poetic text. However, small datasets labeled with a limited number of emotional states are the major limitations of such studies. The existing studies on the detection of emotional states from poetry text have used traditional machine learning techniques with limited datasets tagged with a small number of emotion classes. One of the studies conducted on emotion classification from poetry text has used one machine learning classifier, namely support vector machine and a BiLSTM classifier, for classifying poetry text into two emotion classes.

Emotion can be conveyed in several forms, such as face, movements, voice, written language. Emotion recognition in text is an issue identified based on the principles derived from deep learning and in recent research into human interaction has a vital role in emotional reactions. Nowadays so many people are reading novels and poetry but they do not understand the exact meaning of the lines that are written by the author. So we are trying to convert text into their respective emotion states so that the reader will understand the exact meaning.

II. LITERATURE SURVEY:

[1] Poem is a type of literature designed to convey ideas, emotions, and experiences in a brilliant way. In this article, we discuss the automatic emotion recognition of poems written in English. This is a pioneering approach in emotion recognition from poems. Emotions from the poems, classified into nine emotions, based on 'Navarasa' under 'Rasa Theory' which is described in 'Natya Shastra' written by 'Bharata Muni.' The nine basic emotions such as Love, Sadness, Anger, Hatred, Fear, Surprise, Courage, Joy, and Peace, classified as “Navarasa”. As to our knowledge, we are not familiar with a text corpus of poems based on nine emotions, we have manually created an emotion tagged corpus from poems in English. The corpus created is from an exhaustive collection of poems of Indian poets from the period 1850-1950. The poems are mined from the web, and we applied a ten cross fold Naïve Bayes classifier to recognize the emotion of a poem by maximum likelihood probability.
[2] The country represents the prosperity of that country. India, being a multilingual country, is having a rich heritage and literature. In order to retrieve literature pieces easily, it must be classified. In this research work, vocabulary-content based classification of Punjabi poetry is done. 4 different poetry categories are populated with 240 poems (with 60 poems in each category). These 240 poetry documents are passed through typical NLP text classification phases like Sentence Splitting, Tokenization and Bag-of-Words (BOW) representation, finally yielding to their Vector Space Model (VSM) representation. Total 9867 unique words extracted from last step are used for building the different machine learning models. For the first time in research community, 10 different machine learning algorithms are trained and tested for any Indian language, using weka, with an aim to find the most suitable algorithm.

[3] The poor languages, like Odia, inherently lack the necessary resources and tools for the task of sentiment analysis to give promising results. With more user-generated raw data readily available today, it is of prime importance to have annotated corpora from various domains. This paper is a first attempt towards building an annotated corpus of Odia poetry with sentiment labels. The annotated corpus is further used for sentiment classification using machine learning techniques in order to establish a baseline. Stylistic variations and structural differences between poetic and non-poetic texts make the task of sentiment classification challenging for the former. Using the annotated corpus of poems, we obtained comparable accuracies across various classification models. Linear-SVM outperformed other classifiers with a macro F1-Score of 0.68. The annotated corpus contains a total of 730 Odia Poems of various genres with a vocabulary of more than 23k words. Fleiss Kappa score of 0.83 was obtained which corresponds to near perfect agreement among the annotators.

[4] Although sentiment analysis in Chinese social media has attracted a lot of interest in recent years, it has been less explored in traditional Chinese literature (e.g., classical Chinese poetry) due to the lack of sentiment lexicon resources. In this paper, we propose a weakly supervised approach based on Weighted Personalized PageRank (WPPR) to create a sentiment lexicon for classical Chinese poetry. We evaluate our lexicon intrinsically and extrinsically. We show that our graph based approach outperforms a previous well-known PMI-based approach (Turney and Littman, 2003) on both evaluation settings. On the basis of our sentiment lexicon, we analyze sentiment in the Complete Anthology of Tang Poetry. We extract topics associated with positive (negative) sentiment using a position-aware sentiment topic model. We further compare sentiment among different poets in Tang Dynasty (AD 618 - 907)

[5] This report summarizes the objectives and evaluation of the Sem Eval 2015 task on the sentiment analysis of figurative language on Twitter. This is the first sentiment analysis task wholly dedicated to analyzing figurative language on Twitter. Specifically, three broad classes of figurative language are considered: irony, sarcasm and metaphor. Gold standard sets of 8000 training tweets and 4000 test tweets were annotated using workers on the crowdsourcing platform Crowd Flower. Participating systems were required to provide a fine-grained sentiment score on an 11-point scale (-5 to +5, including 0 for neutral intent) for each tweet, and systems were evaluated against the gold standard using both a Cosine similarity and a Mean-Squared-Error measure.

III. METHODOLOGY:

3.2: The LSTM Algorithm

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in the following work. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn! All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. Step 1: The data is taken in the foCollect a dataset of images of healthy plants and plants with nutrient deficiencies. We can use images from public datasets or create our own dataset by taking pictures of plants with different types of nutrient deficiencies. Label the images in our dataset to indicate which plants are healthy and which have nutrient deficiencies. Train a deep learning model on our labelled dataset. You can use a pre-trained model like ANN, CNN, INCEPTIONRESNET, DENSNET, VGG, and MOBILENET as a starting point, and fine-tune it on your dataset. For getting better accuracy we prefer MOBILENET.
The main aim of the study is to convert the text into their respective emotional states using emotions like joy, fear, anger, sadness, and neural. Nowadays so many people are reading novels and poetry but they do not understand the exact meaning of the lines that are written by the author. So we are trying to convert text into their respective emotion states so that the reader will understand the exact meaning. This can be done by using Long Short-Term Memory (LSTM) which is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.

**Create Dataset:**

The data is taken in the form of csv file. (data.csv)

**Pre-processing:**

Resizing and reshaping the text into appropriate format to train our model

**Training:**

Use the pre-processed training dataset is used to train our model using Deep learning and machine learning algorithms along with Bilstm, RNN.

**Classification:**

The texts are classified according to their emotional states. They are classified as neutral, joy, sadness, fear, love. Based on the text that is given it is classified into respective emotional states.

### 3.2 Algorithm for classification of poetry text into emotional states

**Step 1:** The data is taken in the form of csv file. (data.csv)

**Step 2:** After the input dataset is given, the data will be preprocessed by · Removing Null values from a data frame and replace NaN values with default values. · Sometimes our data will be qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to understand texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data. · Then it fit the model to the data, then transform the data according to the fitted model.

**Step 3:** After the preprocessing, the data is scaled to a fixed range - usually 0 to 1. The cost of having this bounded range - in contrast to standardization - is that we will end up with smaller standard deviations, which can suppress the effect of outliers. Then using s_to_super function the first column of row(t) is shifted to last column of row(t-1) and concatenated. This act transforms a normal preprocessed dataset to recurrent dataset.

**Step 4:** Now we need to split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e. our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task, we will import test_train_split from model_selection library of scikit.
Step 5: Now to build our training and test sets, we will create 4 sets—X_train (training part of the matrix of features), X_test (test part of the matrix of features), Y_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices), Y_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices). We will assign to them the test_train_split, which takes the parameters — arrays (X and Y), test_size.

Step 6: Now, we need to build a model to train the data. Here the model used is Long ShortTerm Memory.

Step 7: An LSTM has a similar control flow as a recurrent neural network. It processes data passing on information as it propagates forward. The differences are the operations within the LSTM’s cells. These operations are used to allow the LSTM to keep or forget information.

Step 8: The first step in LSTM is to decide what information you are going to throw away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It gives a value between 0 and 1, where a 1 represents “keep this as it is” while a 0 represents “get rid of this.” Fig 5.5 Forget Gate This step has two parts: First, a sigmoid layer called the “input gate layer” decides which values we’ll update. Next, a tanh layer creates a vector of new candidate values that could be added to the state. In the next step, by combining these two layers, a new update is being created. 23 Fig 5.6 Input Gate

Step 9: It is now time to update the old cell state, Ct−1, into the new cell state Ct. The last step has already created an update. We only need to update it. Fig 5.7 current state

Step 10: Finally, we need to decide what we’re going to output based on the context that we have selected. Fig 5.8 output layer

IV. Experimental Results and Analysis:

4.1 Libraries used

In this project, we are using python language to develop the project and we need to import the following libraries which are to be used for further processing the input and output.

a) NumPy:

The fundamental Python library for scientific computing is called NumPy. A multidimensional array object, various derived objects (like masked arrays and matrices), and a variety of routines for quick operations on arrays are provided by this Python library. These operations include discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more. The nd-array object is at the centre of the NumPy package.

b) Pandas:

Pandas is an open-source Python library that provides high-performance capabilities for data manipulation and analysis. The term "Pandas" refers to a multidimensional data econometric technique called "Panel Data." Many academic and professional fields, including finance, economics, statistics, analytics, etc., use Python with Pandas.

c) Matplotlib:

Matplotlib is an excellent Python visualisation package for 2D array displays. To handle the larger SciPy stack, a multi-platform data visualisation toolkit called Matplotlib was developed and is based on NumPy arrays. John Hunter introduced it for the first time in 2002. One of visualization's main benefits is that it allows us visual access to enormous amounts of data in easily understandable forms. In Matplotlib, there are many different types of plots, such as line, bar, scatter, and histograms.

4.2 Models Used

In this project, we use different models for training like RNN,BILSTM. Here is an overview of how these different deep learning models could be used for the task of identifying emotional states.

Recurrent Neural Networks (RNN) are a type of deep learning model that can be used is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

Long Short Term Memory(LSTM) networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in the following work. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.
Deep learning method is gradually developing as a promising technique for text classification. In this project, we used different deep learning models for the classification of poetry text into emotional states. Here we trained LSTM network on text classification so that it converts text into respective emotional states. So, the readers will understand the exact meaning of the text that author want to convey. The analysis can be further extended by utilizing methods like Convolution Neural Network to classify the text into emotional states.

4.3 Web application for identification of emotional states

RNN

By using this model, we identified that the leaf has Magnesium Deficiencies with an Accuracy of 90.909.

ANN

Fig. 4.2.1: Analysis of different models

Fig 8.1 Dataset for text data
By using this model, we identified that the leaf has Magnesium Deficiencies with an Accuracy of 84.091.

```python
data_train = pd.read_csv('data/data_train.csv', encoding='utf-8')
data_test = pd.read_csv('data/data_test.csv', encoding='utf-8')

X_train = data_train.Text
X_test = data_test.Text

y_train = data_train.Emotion
y_test = data_test.Emotion

data = data_train.append(data_test, ignore_index=True)

print(data.Emotion.value_counts())
data.head(5)
```

**Fig 8.2 Data Splitting for text data**

**DENSENET**

By using this model, we identified that the leaf has Magnesium Deficiencies with an Accuracy of 95.864.

![Model Accuracy Chart](image_url)

**Fig 8.3 Model Accuracy for text data**
INCEPTION RESNET

By using this model, we identified that the leaf has Magnesium Deficiencies with an Accuracy of 94.318.

![Fig 8.4 Model Loss for text data](image1)

MOBIENET

By using this model, we identified that the leaf has Magnesium Deficiencies with an Accuracy of 97.727.

![Fig 8.5 Confusion Matrix for text data](image2)
V. CONCLUSION:

Deep learning method is gradually developing as a promising technique for text classification. In this project, we used different deep learning models for the classification of poetry text into emotional states. Here we trained LSTM network on text classification so that it converts text into respective emotional states. so, the readers will understand the exact meaning of the text that the author wants to convey. The analysis can be further extended by utilizing methods like Convolution Neural Network to classify the text into emotional states.

REFERENCES:


