Arabic tweets and news classification

Muhammad Ayaz Khan

GitHub link https://github.com/muhammadayazkhanuet/arabic-tweet-and-news-classification

Introduction

Unstructured data is everywhere social media, emails, chats, webpages and much more. Text classification is the task of assigning a set of predefined categories to open-ended text. It helps in organizing, structuring and categorizing any kind of textual information weather medical, general files or the whole web. Classifying text is the main task of natural language processing (NLP). Depending on the content of the text strings classifying it into different categories is called natural language processing. The application of text classification includes sentiment analysis from the text or speech classifying a tweet as spam or no spam and automatically classifying blog posts into corresponding categories.

Research work

In this work we have worked on the Arabic tweet classification and Arabic news classification using machine learning techniques. we have taken two Arabic data sets. On data set [1] comprised of 57000 samples of negative and positive sentences. We have trained the model which will predict the given input sample to either positive or negative class or category or sentiment. Another work is the categorization or classification of text or news. Wither a certain input text or news belongs to which category/class of news. Dataset consists of 10360 samples. It includes local news, international news, sports and cultural news. The model will then specify about the input whether it belongs to which of the four classes of news. The results are shown in the results part of this document.

Working of text classification

By using pre-labeled examples as training data, machine learning algorithms can learn the different associations between pieces of text, and that a output (i.e., tags) is expected for a particular input (i.e., text). A “tag” is the pre-determined classification or category that any given text could fall into. The first step towards training a machine learning NLP classifier is feature extraction: a method is used to transform each text into a numerical representation in the form of a vector. One of the most frequently used approaches is bag of words, where a vector represents the frequency of a word in a predefined dictionary of words. For example, if we have defined our dictionary to have the following words {This, is, the, not, awesome, bad, basketball}, and we wanted to vectorize the text “This is awesome,” we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0). Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. sports, politics) to produce a classification model. Text classification with machine learning is usually much more accurate than human-crafted rule systems, especially on complex NLP classification tasks. Also, classifiers with machine learning are easier to maintain and you can always tag new examples to learn new tasks.

Sentiment Analysis

One of the popular examples of text classification is sentiment analysis. It is the process of reading text and knowing the polarity of text weather positive, negative, neutral or some other. Products monitoring, brand analytics and market research are the areas where sentiment analysis are used. Sentiment analysis allows you to automatically analyze all forms of text for the feeling and emotion of the writer. It helps in knowing the feeling of the writer from the text or the post or the tweet.

Intent Detection

Intent detection or intent classification is another great use case for text classification that analyzes text to understand the reason behind feedback. Maybe it’s a complaint, or maybe a customer is expressing intent to purchase a product. It’s used for customer service, marketing email responses, generating product analytics, and automating business practices. Intent detection with machine learning can read emails and chatbot conversations and automatically route them to the correct department.

Tweet classification model

Dataset

The data set used in the work is taken from Kaggle [1]. This dataset consists of two classes of text i.e positive and negative sentence represented with emojis as well. The dataset consists of a 58749 number of samples. In this dataset the negative samples/sentences are 28901 and positive samples/sentences are 29848. The batch size is set to 32. The training and validation ratio are 0.2.

Model

The keras sequential model is used our work. In our model the first layer is the embedding layer with its input parameters and then two bidirectional layers with its input parameters. The fourth layers is the dense layer with 64 and with reLu activation function. The fifth layer is the dropout layer with 0.5 and the last layer is the dense layer with 1 as input to define two classes of either positive or negative sentence or sample.

Loss function

Loss functions is used for model optimization here cross entropy loss function is used. Each predicted class probability is compared to the actual class desired output 0 or 1 and a score/loss is calculated that penalizes the probability based on how far it is from the actual expected value. The penalty is logarithmic in nature yielding a large score for large differences close to 1 and small score for small differences tending to 0. Cross-entropy loss is used when adjusting model weights during training. The aim is to minimize the loss, i.e, the smaller the loss the better the model. A perfect model has a cross-entropy loss of 0.

Optimizer

The Adam optimizer is used in this work. Adam optimizer is used for training deep neural networks in replacement of stochastic gradient descent method. It combines AdaGrad and RMSProp like alogorithms for optimization. It can handle sparse gradients on noisy problems.

Results

In this work the batch size is set to 32 and the number of epochs is set to 10. The results of the first epoch are as below. In the first epoch the loss is 0.514, the accuracy is 0.6623 and validation accuracy is 0.9198. in the 10th epoch the loss is 0. 0401, the accuracy is 0.9865. the validation accuracy is 0.9260 and validation accuracy is

0.9198. and validation accuracy is 0.9198. the test loss is 0.08708123117685318 and the test accuracy is 0.9768174290657043.

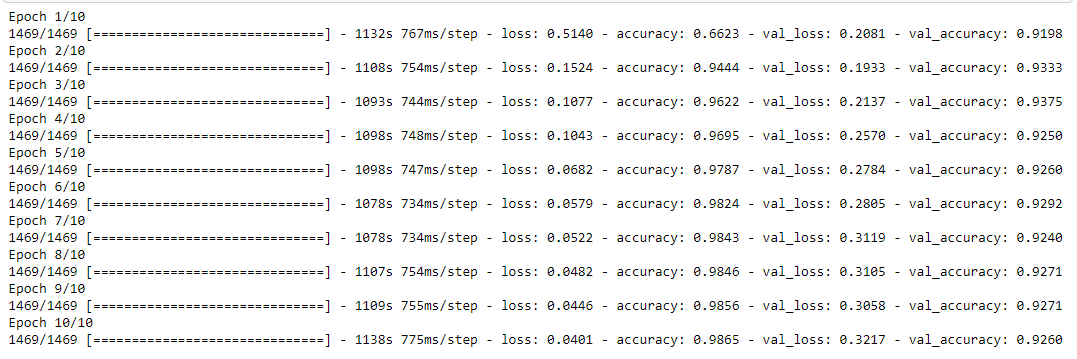


Fig1.1 Results over ten epochs

In the graph we have show the training and validation accuracy and loss both differently. The training accuracy increases as the number of epochs increase till it somehow becomes constant after 10 epochs. The validation accuracy has variations in the start by increasing then decreasing and then becomes constant as show in the graph. The variation of the training loss is obvious and more the number of epochs and same with the validation loss shown here.

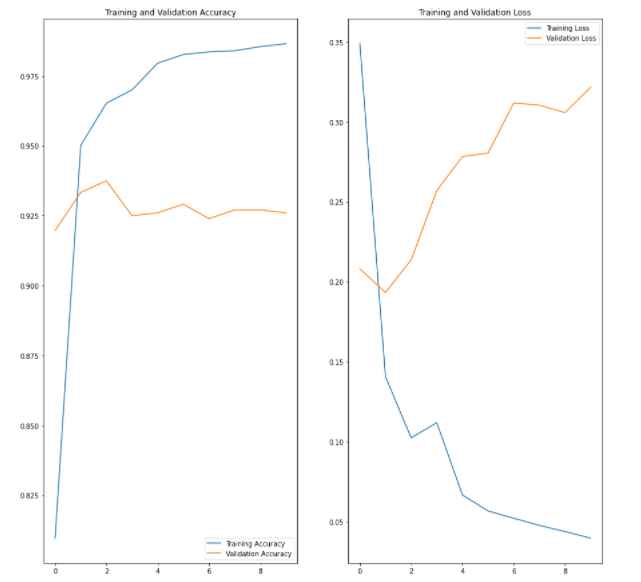


Fig1.2 Training, validation accuracy and loss

News classification model

This model uses embedding layer. For pooling the global average pooling mechanism is used. A dense layer with N\_CLASSES is used for the network. For lose function the categorical cross entropy and for optimization adam optimizer is used. Here we have kept the batch size of 128 and the number of epochs are set to 40. The training and validation ratio is 0.2.

Results for news classification model

In the first epoch the loss function gives 2.1476 and the accuracy is 0.3347 while the validation loss and validation accuracies are 1.9824 and 0.4653 respectively. In the last epoch the loss function gives 0.4021 and the accuracy

References

[1] [https://www.google.com/search?q=arabic-sentiment-twitter- corpus&oq=arabic-sentiment-](https://www.google.com/search?q=arabic-sentiment-twitter-%20%20%20%20%20%20%20corpus&oq=arabic-sentiment-) twitter-corpus&aqs=chrome.0.69i59j69i60l3.352j0j7&sourceid=chrome&ie=UTF-8

[2 https://github.com/muhammadayazkhanuet/arabic-tweet-and-news-classification