Team MohammadHabash at Mowjaz Multi-Topic Labelling Task

Mohammad Habash
Dept. Computer Science
Jordan Univ. of Science and Technology
Amman, Jordan
mshabash187@cit.just.edu.jo

***Abstract*—Multi-label text classification is an important problem with the growing size of data and the difficulties in assigning a single label to each text sample because of the tendency of internet users to assign multiple labels to describe documents, emails, posts, etc. Our goal is to predict the category (topic) of an article given its text. The dataset which is used in this work contains articles from Mowjaz. Mowjaz is an Arabic topical content aggregation mobile application for news, sport, entertainment and other topics from top publishers that users can follow. This paper describes the approach to classify articles using Bi-directional Gated Recurrent Unit (Bi-GRU) with AraVec embeddings. The F1-score of this system is 0.8344 which shows a significant improvement over the baseline models.**

Keywords—multi-label classification; machine learning; deep learning; Gated Recurrent Unit; Arabic text; RNN;

I. INTRODUCTION

This paper describes my contribution to ICICS 2021 Mowjaz Multi-Topic Labelling Task [1]. In machine learning and statistics, classification is defined as training a system with labeled dataset to identify a new unseen dataset to which class it belongs. In multi-label classification, one sample can belong to more than one class. Arabic language is considered difficult to deal with. This is because Arabic morphology is so complex and there exists variety of dialects in the Arabic language. Most of the existing works focus on English text. This work focuses on classifying Arabic articles, which are provided by Mowjaz, and describes the approach to classify each article to its category (possibly more than one category) using Bi-directional Gated Recurrent Unit (Bi-GRU).

II. METHODOLOGY

A. Dataset

The dataset consists of 9,590 articles split into training, development and testing sets. The following table represents some statistical properties of this dataset.

TABLE I. DATASET STATISTICS

	Training	**Development**	**Testing**
Articles No.	7,681	956	953
Max/Min Article length	3,384/1	2,418/1	1,602/2
Avg/StdDev article length	228.6/218.2	226.2/227.5	235.1/216.0
Number of unique words	228,765	55,256	57,853

The following figure shows the number of articles in each category.

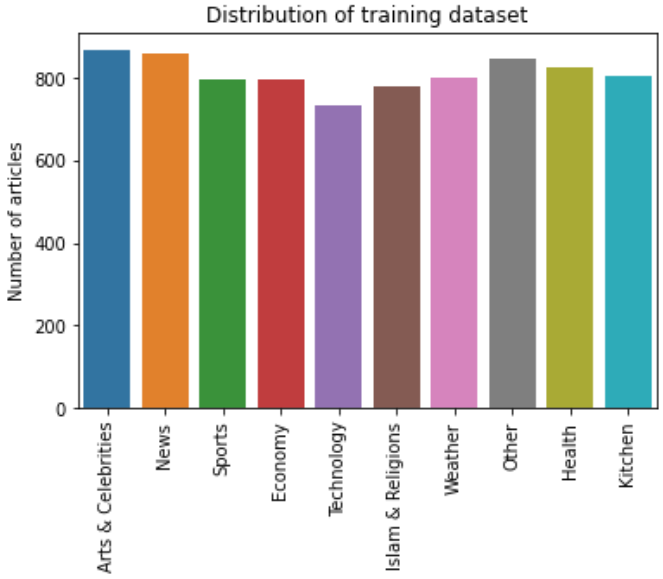

Fig. 1. Distribution of training dataset

Fig. 1. shows that the training dataset distribution is almost perfectly balanced. No actions are needed to make the dataset more balanced. In other words, we do not have to apply an **under-sampling** or **over-sampling** method.

B. Model

Dataset contains both articles and their labels, so we are dealing with a supervised learning problem. Deep neural networks (DNN) have recently shown significant improvements over traditional Machine Learning (ML) based approaches on classification tasks. For example, Convolution Neural Networks (CNN) and Recurrent Neural Networks (RNN) outperform traditional machine learning approaches, such as Support Vector Machine (SVM). Long-Short Term Memory Network (LSTM) is a type of RNN that uses special units in addition to standard units. These special units can 'memorize' longer sequences. Gated Recurrent Unit (GRU) controls the flow of information like LSTM, but without having to use a memory unit. Both LSTM and GRU prevent vanishing gradient problem, which exists in a standard RNN. After experiments on our dataset, **GRU seems to perform as good as LSTM**. Because GRU is less complex and has less parameters, it is the Neural Network chosen in this classification approach. We will be using Bi-directional GRU (Bi-GRU), which allows to extract features from the original (forward) sequence and reversed sequence. More details about the model will be presented in section IV.

III. TEXT PREPROCESSING AND WORD REPRESENTATION

Arabic and English punctuation (including parentheses, underscores, quotes, etc.) are removed from all articles in the training, development, and testing sets. As well as, html tags, web addresses, twitter usernames. For Arabic words, tashkeel and tatweel are stripped using pyarabic [3]. Words of length less than 3, non-Arabic words (English, Unicode, etc.) and numbers are also removed. Arabic stop-words are removed to minimize sequences' lengths. The preprocessing effect is shown in the following table, which displays a sequence before and after the preprocess.

TABLE II. TEXT PREPROCESSING EXAMPLE

	Article
Before Preprocessing	عمان 4 كانون الأول (بترا) -<p> تكون الأجواء اليوم باردة نسبياً وغائمة جزئياً في أغلب المناطق، ولطيفة الحرارة في الأغوار والبحر الميت، وتكون الرياح شمالية غربية خفيفة السرعة،
After Preprocessing	عمان الاول بترا الاجواء بارده نسبيا وغائمه جزئيا اغلب المناطق ولطيفه الحراره الاغوار والبحر الميت وتكون الرياح شماليه غربيه خفيفه السرعه

Using python "WordCloud" library, we are going to be drawing a word cloud of all articles in training set.

Fig. 2. Word Cloud of Mowjaz training set articles

Now after our articles are almost clean and noiseless, we can represent the words in articles as a list of numeric values that can be fed to a neural network, by applying word embedding technique. We will be using AraVec [2], which is a pre-trained word embedding language model. We will be using a unigram, skip-gram model built on Arabic tweets, with a vector size of 300, which means that every word will be represented by a list of 300 numbers, and a maximum sequence length of 256 is used. If a sequence's length is less than 256, **the sequence is padded with zeros**. This is done to obtain a constant input shape of (256, 300) to the model.

IV. DEEP LEARNING MODEL

The Sequential model takes the embedded sentence as an input, with shape (256, 300), followed by a Dropout layer with a drop rate of 0.2. The Dropout layer is a regularization technique which randomly sets input units to 0 at each step during training time, which helps prevent overfitting. Next, one Bi-directional Gated Recurrent Unit (Bi-GRU) of 300 cells for each of the forward layer and backward layer. Finally, a Dense layer (fully connected layer) of 300 neurons. At the end, a Dense layer of 10 neurons to represent each article category as an output. Total number of parameters of the model is **1,266,910**.

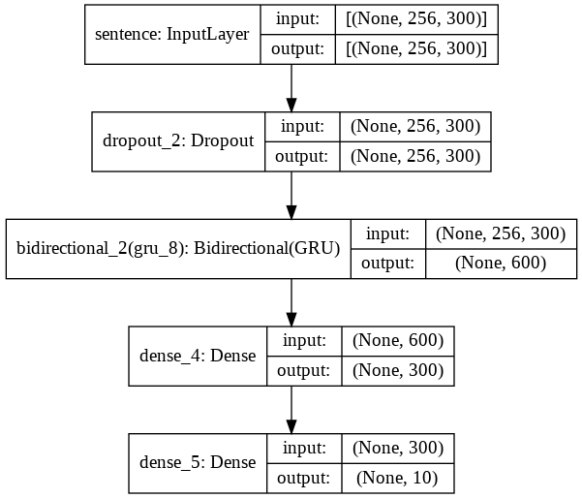

Fig. 3. Model Architecture

Model is compiled using Adam optimizer with a learning rate of 0.0005, training batch size of 50, 8 epochs and 300 word embedding size as the hyperparameters. Adam optimizer with a learning rate of 0.0005 was proper and slow enough to converge. It was faster and more efficient than other optimizers like SGD, RMSprop, and it achieved better results.

TABLE III. MODEL HYPERPARAMETERS

Optimizer	**Learning rate**	**Batch size**	**Epochs**	**Embed. Size**
Adam	0.0005	50	8	300

V. RESULTS

To measure the accuracy of our model, F1 score is used. It is a good choice to test the model as it conveys the balance between precision and recall. F1 score results for validation set with a 0.3 classification threshold:

- F1 macro: 0.865
- F1 micro: 0.861
- F1 samples: 0.872

Model's accuracy on the test set is **0.8344**.

VI. CONCLUSION

In this paper, I introduced the approach to classify Mowjaz articles using Bi-GRU. The dataset was well balanced and there was no need to take actions to balance it. There was much noise in the articles' text. Text cleaning (preprocessing) was necessary to make the articles noiseless. After experiments, GRU and LSTM resulted in approximately the same accuracy. GRU was chosen because it is less complex. Bi-GRU performs better than a standard GRU as it extracts features of the reversed sequence. I tried other deep learning models, such as CNN, but they did not excel Bi-GRU. I found that Adam optimizer was the best optimizer for this task. Increasing the model's complexity (more layers and neurons) did not achieve better results, instead, in most times it caused overfitting.

REFERENCES

[1] M. Al-Ayyoub, H. Selawi, M. Zaghlol, H. Al-Natsheh, S. Suileman, A. Fadel, R. Badawi, A. Morsy, I. Tuffaha, and M. Aljarrah, "Overview of the Mowjaz Multi-Topic Labelling Task," in The 12th International Conference on Information and Communication Systems (ICICS 2021).

[2] Abu Bakr Soliman, Kareem Eisa, and Samhaa R. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP", in proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017), Dubai, UAE, 2017.

[3] T. Zerrouki, Pyarabic, An Arabic language library for Python, https://pypi.python.org/pypi/pyarabic, 2010.

[4] Francois Chollet et al. Keras. https://keras.io, 2015