Deep learning vs Machine Learning in Arabic Text Classification

Natural Language Processing and Information Retrieval Project

1st Malik Ziq
dept. Electrical and Computer Engineering
Birzeit University
Birzeit, Palestine
malikziq1@gmail.com

2nd Hadi Jaradat
dept. Electrical and Computer Engineering
Birzeit University
Birzeit, Palestine
hadijaradat@gmail.com

***Abstract*—This project aims to compare between Deep Learning and classical Machine Learning models on text classification problem using an Arabic text corpus, and Natural Language Processing to clean the text and represent it in a way that computer can deal with. The comparison were between three models, Deep Neural network, LSTM with word Embedding and SVM model with Bag of Words and TF-IDF for feature extraction and text representation.**

***Index Terms*—Natural Language Processing, Arabic, Deep Learning, Machine Learning**

I. Introduction

Solving classification problem depends highly on the data-set and the problem. The classifier can be made using two main techniques, Deep Learning or classical Machine Learning. This project, uses an Arabic corpus that has five different categories with different records in each class. Where this corpus used to train Deep Learning neural network and Long Short-Term Memory (LSTM) model with Word Embedding.

Also, to train Support Vector Machine (SVM) with Bag of words (BOW) and TF-IDF, after natural language processing and cleaning the corpus. The TF-IDf was used to give more insite on the relation of the words in the document and on the overall corpus.

II. Related Work

A. Natural Language Processing (NLP)

NLP is the process of converting text from humane language to a representation that can be understood by computers. Since computers can't understand the natural language which is the language, that humans uses to communicate with each other. [1]

Some of these techniques are:

*1) **Word Embedding**:* Word Embedding is a process of representing words in a text in away where words that are close in meaning will have close representations. Word embedding is used in Deep Learning as a Embedding Layer, which is a learned neural network. this layer takes the text represented as one-hot encoded which means a one if a words exists and zero if not, to learn and produce the vector that will represent the text in a better way, since it takes the relations between words in case. [2]

*2) **Bag of Words (BOW)**:* Another method of representing text is Bag of Words, where the text represented as the frequency of each word in it.

*3) **Term Frequency-Inverse Document Frequency (TF-IDF)**:* This representation consist of the multiplication of two main components: Term Frequency (TF) which is the frequency of the word in a document or a text, and Inverse Document Frequency which is the inverse count of the words in all documents

$$idf = \log \frac{N}{tdf} \tag{1}$$

Where, N is the total number of documents and tdf is the term frequency in all documents. the main benefit of this representation is that the most frequent words will have low wight. [3]

B. Deep Learning

Shortly, Deep learning is a multiple layers of neural networks that can be connected in different way to achieve a task of making a decision, from learning and computing the weights using the data-set. One of the well known neural networks that is used in text classifiers is the Long Short-Term Memory neural network, this neural network has a feed back connection which makes it better in connecting words representations in the same context. [4]

C. Machine Learning

Machine learning divides into two areas supervised where that data have labels (e.g. the class, category, etc.) or unsupervised where the data doesn't have labels. SVM, is one of the most common supervised learning algorithms that is used in classification problems. SVM takes the data points and produce the best hyperplane, this hyperplane will be the best separator of the data points, that improves over learning and training on data. [5]

III. Data Set

The data-set used in this project is an Arabic Text corpus for text classification, a collection of texts collected from newspapers Assabah [6], Hespress [7] and Akhbarona [8]. [9] The data set is labeled into five categories as in table I.

TABLE I
DATA CATEGORIES

Category	*ID*	*Total Records*
Culture	0	13738
Divers	1	16728
Economy	2	14235
Politic	3	20505
Sport	4	46522

Moreover, the data-set has:

- 314835 tokens.
- Average text length of 242.4 words.
- Max text length of 4279 words.

TABLE II
WORDS FREQUENCY IN THE CORPUS

Word	*Rank*	*Frequency*
في	1	741590
من	2	578526
على	3	344821
أن	4	331433
إلى	5	291822
التي	6	179403
الذي	7	138182
عن	8	127907
مع	9	104563
ما	10	93607

From table II, the top ten most frequently words in the corpus are all stop words. That is because of the heavy using of these stop words in all documents and texts.
Next is Zipf's graph for the data-set of the top 20 words.

Fig. 1. Zipf's Law Graph.

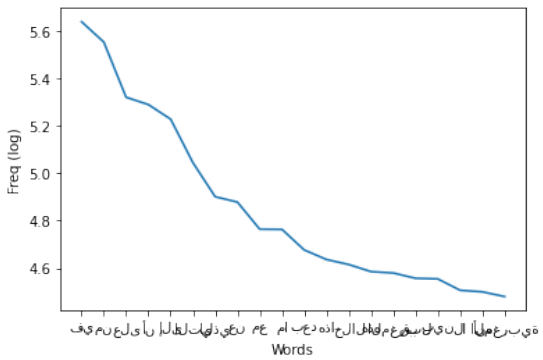


Figure 1 shows the log of the frequency vs word. Obtain that the relation of Zipf's Law between the word frequencies and their ranks is satisfied.

IV. Deep Learning Implementation

The Deep learning Implementation is divided into the following stages, where two models were implemented a deep neural network and LSTM:

A. *Pre-processing Data*

The first stage is to get the data ready for the models, since the data in text format. This stage is shared between the two models. The data was split into 80% training and 20% testing. Then the training and testing data was tokenized using Keras Tokenizer to represent the data in a numerical format, from text to a vector that has the words ranks in the corpus. As in the following example:

Text:
من الذين على الأدبية يحل الشاعر
Vector:
[[2, 80, 3, 7919, 2942, 4076]]

In the above example, the text from right to left is converted to the vector in the list from left to right. Where each word is represented by its rank in the corpus.

Next, the labels was categorized and one-hot encoded using Keras. And the texts' vectors was padded since the max sequence length parameter in the model was set to 300, the padding was done using Keras.

B. *Deep Neural Model*

The First model consist of the the layers as in fig 2:

Fig. 2. Deep neural model layers.

```
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 300)               0
_________________________________________________________________
embedding_3 (Embedding)      (None, 300, 50)           1000000
_________________________________________________________________
global_average_pooling1d_3 ( (None, 50)                0
_________________________________________________________________
dense_3 (Dense)              (None, 5)                 255
=================================================================
Total params: 1,000,255
Trainable params: 1,000,255
Non-trainable params: 0
_________________________________________________________________
```

Note In fig 2, the first layer is the input layer with 300 neurals connected to the embedding layer. The embedding layer is the layer that will represent the text vector as a 300 x 50, where 50 is the embedding dimension for each of the 300 feature. Finally, is a pooling layer that will do the averaging, and the output layer is dense layer with Softmax activation. The model training parameters:

- 15 Epoch.
- 128 Batch size.
- 80% Training and 20% Validation.
- Adam Optimizer.

Fig. 3. Training Accuracy.

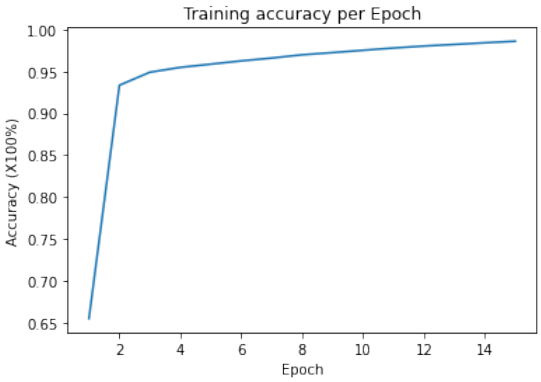


Fig. 4. Validation Accuracy.

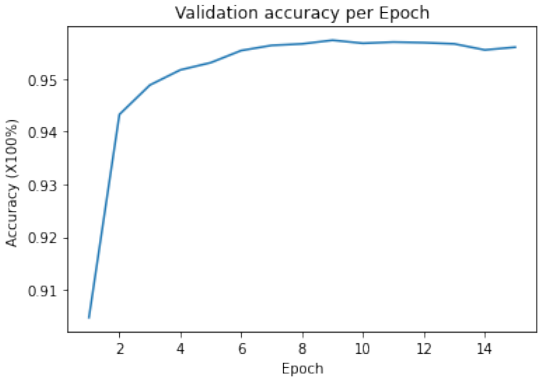
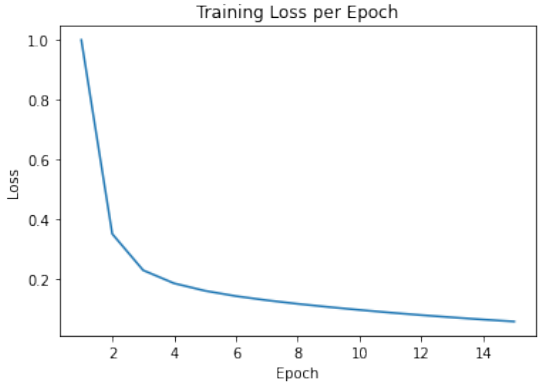


Fig. 5. Training Loss.



The final accuracy of the training is 98.6%, validation is 95.6% and on the testing data is **99.5%**.

C. LSTM

The second model consists of much complected layers than the first model it contains a LSTM layer to test the effect of the LSTM layer on the same data. The data preprocessing and training parameters were as in the first model. Fig 6 shows the model layers:

Fig. 6. LSTM model layers.

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	(None, 300)	0
embedding_3 (Embedding)	(None, 300, 50)	1000000
conv1d_3 (Conv1D)	(None, 296, 64)	16064
max_pooling1d_3 (MaxPooling1	(None, 59, 64)	0
dropout_3 (Dropout)	(None, 59, 64)	0
conv1d_4 (Conv1D)	(None, 55, 64)	20544
max_pooling1d_4 (MaxPooling1	(None, 11, 64)	0
dropout_4 (Dropout)	(None, 11, 64)	0
lstm_2 (LSTM)	(None, 32)	12416
dense_5 (Dense)	(None, 5)	165

Total params: 1,049,189
Trainable params: 1,049,189
Non-trainable params: 0

following is the training and testing graphs:

Fig. 7. Training Accuracy.

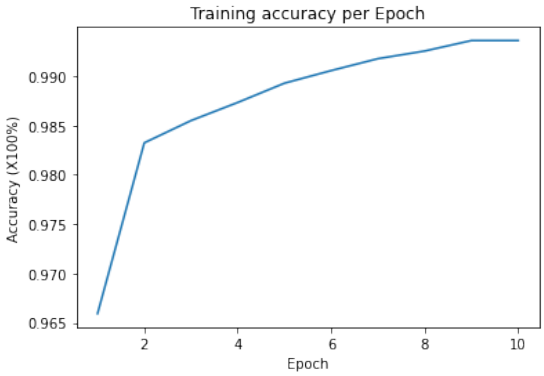


Fig. 8. Validation Accuracy.

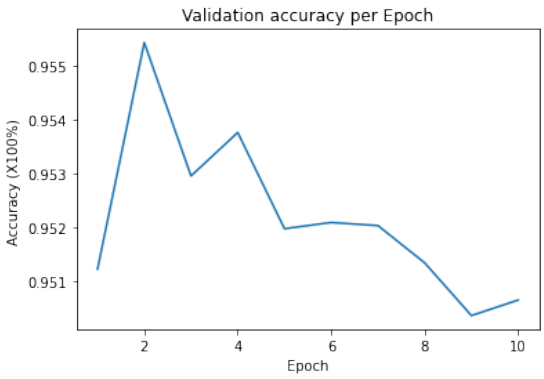


Fig. 9. Training Loss.

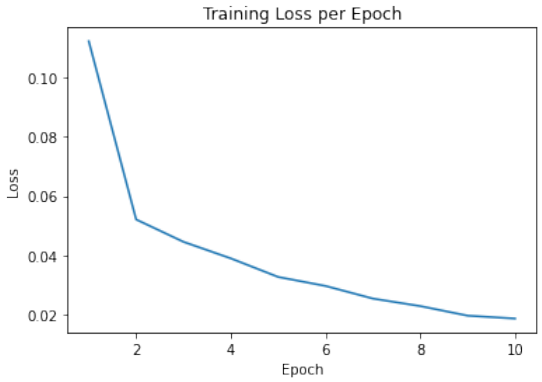
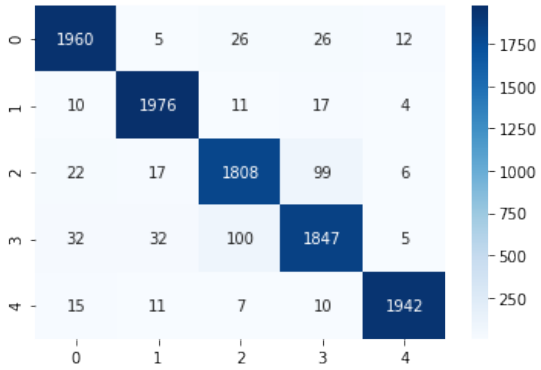


Fig. 10. SVM confusion matrix.



The final accuracy of the training is 99.7%, validation is 94.8% and on the testing data is **99.3%**.

V. Machine Learning - SMV Implementation

The stages for the Machine Learning is a bit different since Machine learning model like SMV needs a feature vector as input in this project Bag of Words and TF-IDF are used to represent the feature vector of the input text. Also, before feature extractions a preprocessing on the data is needed.

Following is the process taken in this implementation:

A. *Pre-processing*

Since the stop wards in the text are in all documents this makes them less uesful, so all the stop words was removed using NLTK library. Also, to minimize number of tokens the ISRI stammer as used. Bellow is an example on the stop words removal and stemming:

Before:

كبير في ميدان العلم والمعرفة لدى المغاربة والمسلمين بصفة عامة

After:

كبر ميد علم عرف غرب سلم بصف عمة

B. *BOW and TF-IDF*

After data preprocessing the bag of words algorithm was used with max words feature of **5000 words** in the dictionary, then the TF-IDF was calculated for each feature in the feature vector using Scikit Learn APIs.

C. *SVM*

Finally, the feature vectors was used to train the SVM model after splitting the data to 80% training and 20% testing.

SVM parameters:

- RBF kernel.
- 80% Training and 20% Testing.

The final results of the SVM model are 98.8% training accuracy and **95.3%** testing accuracy.

Following is the confusion matrix:

VI. Conclusion

In this project a Arabic Text corpus for text classification data-set was used to compare between Deep learning and Machine learning models.

For deep learning a deep neural network and an LSTM models were tested. Where the data was fed to the model directly without any NLP, only a preprocessing to convert the data to numerical number by representing a word with a number instead of a text. So, the Embedding layer in the models will extract the feature vector for each input text on a shape of array with values that represent the words relation in the text. In the other hand, the SVM which is a Machine learning model needed a preprocessing for the text and a feature extraction technique. BOW was used to do the feature extraction for the text, along with TF-IDF so words that does exists in certain categories will have higher wights. Before, the feature extraction a preprocessing for the text was applied where stop words were removed and word stemming for the remaining words.

Finally, results show that Deep Neural Network with accuracy of 99.5% where the best result vs 99.3% for LSTM and 95.3% for the SVM. The code can be found in the link [10].

References

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. 76, pp. 2493–2537, 2011. [Online]. Available: http://jmlr.org/papers/v12/collobert11a.html

[2] J. Brownlee. (2017) Word embedding. [Online]. Available: https://machinelearningmastery.com/what-are-word-embeddings/

[3] B. Stecanella. (2019) Term frequency-inverse document frequenc. [Online]. Available: https://monkeylearn.com/blog/what-is-tf-idf/

[4] Oinkina. (2015) Long short-term memory. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[5] B. Stecanella. (2017) Support vector machine. [Online]. Available: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

[6] Assabah. [Online]. Available: https://assabah.ma

[7] Hespress. [Online]. Available: https://www.hespress.com

[8] Akhbarona. [Online]. Available: https://www.akhbarona.com

[9] Dataset for arabic classification. [Online]. Available: https://data.mendeley.com/datasets/v524p5dhpj/2

[10] (2020) Project source code. [Online]. Available: https://github.com/malikziq/arabic-classification-dl-ml/blob/master/Project.ipynb