Sentiment Analysis For Arabic Reviews

Iman Attia, Ahmed Essam
The American University in Cairo, CSCE Department

Abstract

Arabic sentiment analysis using deep learning is a field of research that aims to develop algorithms and models to automatically classify Arabic text based on its sentiment, i.e., whether the text expresses a positive, negative, or neutral sentiment. It is a very crucial problem to solve as it has a variety of applications, such as monitoring social media sentiment, analyzing customer reviews, and more. We aim to focus on the Arabic Customer Reviews Analysis and attempt to solve it.

Introduction

In our project, **the input** is mainly a string that presents a review of a specific book or hotel written in the Arabic language. **The output** is the predicted rating based on the sentiment analysis for the words in the review ranging from 1 (Bad) to 5 (Excellent).

The attached chart (Fig. 1) indicates an approximate number of open-access papers monitoring the arabic datasets in the last five years. From these datasets, **Hotel Arabic-Reviews Dataset (HARD) & Large-Scale Arabic Book Reviews (LABR)** are chosen to be the perfect choices to train our project because they are **domain specific, diverse, and with high quality.**

We are going to choose the Arabic Bidirectional Encoder Representations from the Transformers (AraBERT) Model. It is an Arabic pre-trained language model based on Google's BERT architecture (Fig. 2). AraBERT uses the same BERT-Base config. The details of the different AraBERT Variations Architectures are displayed in the below table (Fig. 3).

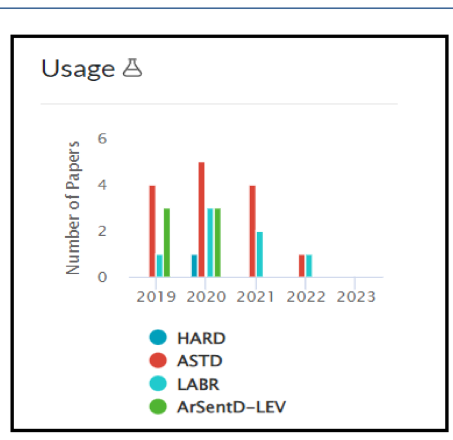


Fig1: Datasets Usage (2019-2023)

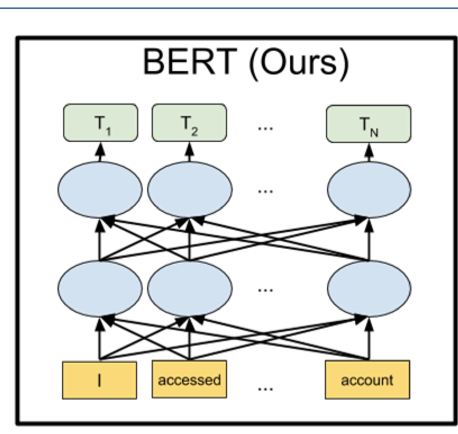


Fig2: BERT Architecture

	AraBert	ArabicBERT Mini	ArabicBERT Medium	ArabicBERT Base	ArabicBERT Large
Hidden Layers	12	4	8	12	24
Attention heads	12	4	8	12	16
Hidden size	768	256	512	768	1024
Parameters	110M	11M	42M	110M	340M

Fig3: AraBERT Model variations

Methods and Experiments

Our methodologies include:

1. Installing Dependencies: used transformers version 4.12 for our model, and we cloned AraBERT model from its github directory to work on: https://github.com/aub-mind/arabert

2. Preparing HARD & LABR datasets & Data Preprocessing: removed unused columns and missing-info rows, concatenated both datasets, and finally applied preprocessing on data using the AraBERT processor.

3. Tokenization: checked the tokenized sentence length to decide the optimal value for the maximum sentence length.

4. Model Deployment: used AutoModelForSequenceClassification.from_pretrained(), passed model name as a parameter.

5. Computing Metrics: We used F1 score, precision, recall, MAE, MSE for our score metrics.

6. Regular Training: trained the model with AraBERT v02 model architecture, using Trainer class optimized for Transformers.

7. Experimenting with K-Fold and Ensemble the Result: used cross validation to search for the best model weights.

8. Fine Tuning the original model: experimented with learning rate, batch size, hidden activation functions, "attention_probs_dropout_prob", "hidden_act", "hidden_dropout_prob", "hidden_size", and "num_attention_heads", "num_hidden_layers".

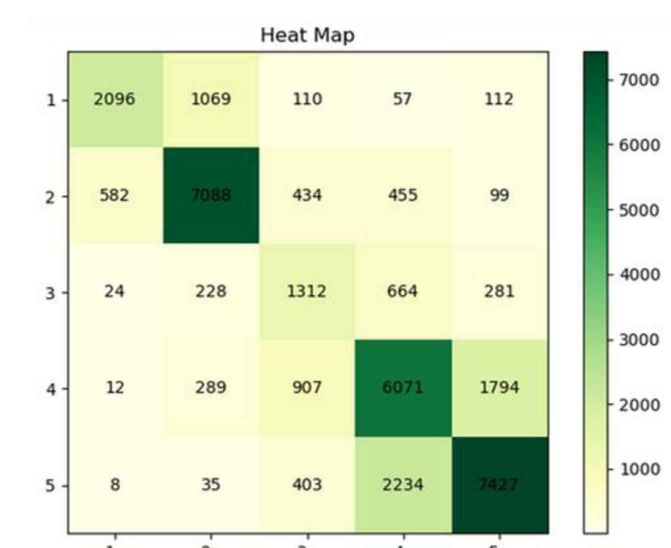
9. Trying Data Augmentation: used Back Translation technique to overcome the class Imbalance for Class 1 and Class 3

10. Integration of Results: Trained the Fine Tuned model on the Augmented Dataset to search for better results.

Training Results

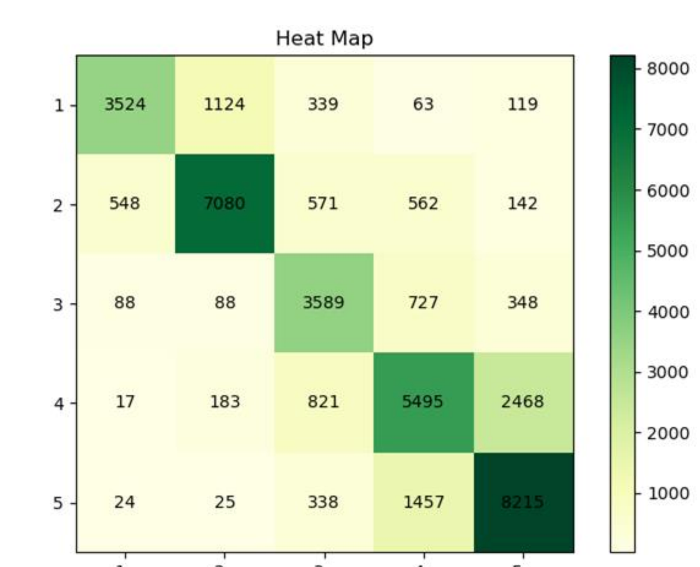
A) Original Model Results on our Concatenated Dataset

Epoch	Training Loss	Validation Loss	Macro F1	Precision	Recall	Mse	Mae
1	0.518200	0.691505	0.672471	0.680655	0.670828	0.529934	0.358823
2	0.490800	0.705915	0.670444	0.683307	0.660791	0.522210	0.350626



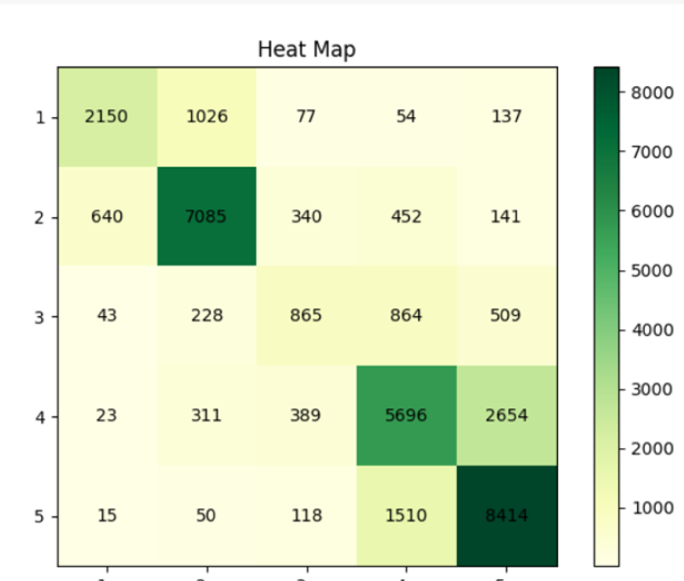
B) Original Model Results on Augmented Dataset

Epoch	Training Loss	Validation Loss	Macro F1	Precision	Recall	Mse	Mae
0	0.617600	0.610479	0.730962	0.739070	0.729370	0.520274	0.338111
1	0.546100	0.596906	0.736808	0.743910	0.731661	0.499038	0.326887



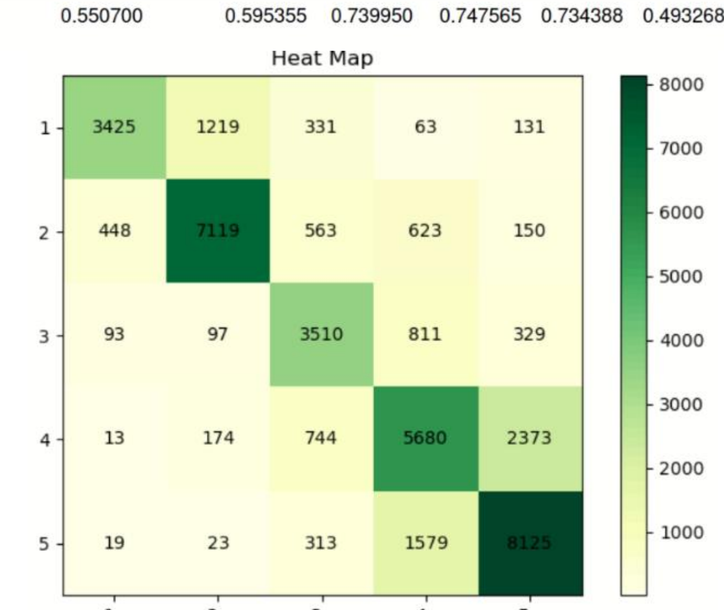
C) Fine-Tuned Model Results on our Concatenated Dataset Dataset

Epoch	Training Loss	Validation Loss	Macro F1	Precision	Recall	Mse	Mae
1	0.650500	0.639980	0.662296	0.684134	0.649528	0.548519	0.357580



D) Fine-Tuned Model Results on our Augmented

Epoch	Training Loss	Validation Loss	Macro F1	Precision	Recall	Mse	Mae
0	0.618100	0.615893	0.729755	0.740859	0.725480	0.525016	0.340061
1	0.550700	0.595355	0.739950	0.747565	0.734388	0.493268	0.323172



Test Cases

Here we will present the test cases for our best performing model:

```
In [84]: # pipe("Some Text")
         pipe("شيء بشع")
Out[84]: [[{'label': 1, 'score': 0.052717216312885284},
           {'label': 2, 'score': 0.07370147109031677},
           {'label': 3, 'score': 0.05697081983089447},
           {'label': 4, 'score': 0.1861371397972107},
           {'label': 5, 'score': 0.6304733157157898}]]

In [85]: pipe("تجربة جميلة!")
Out[85]: [[{'label': 1, 'score': 0.035860348492860794},
           {'label': 2, 'score': 0.050931110978126526},
           {'label': 3, 'score': 0.1310013085603714},
           {'label': 4, 'score': 0.4280247986316681},
           {'label': 5, 'score': 0.3541823923587799}]]

In [86]: pipe("تجربة جميلة جدا لقد سعدت بوجودي في هذا المكان ارشحه و بشده لجميع اصدقائي !")
Out[86]: [[{'label': 1, 'score': 0.0384066142141819},
           {'label': 2, 'score': 0.013043262995779514},
           {'label': 3, 'score': 0.011683846823871136},
           {'label': 4, 'score': 0.1271923929452896},
           {'label': 5, 'score': 0.8096739053726196}]]

In [87]: pipe("تجربة متوسطة لا بأس بها!")
Out[87]: [[{'label': 1, 'score': 0.0449075885117054},
           {'label': 2, 'score': 0.660768985748291},
           {'label': 3, 'score': 0.11075500398874283},
           {'label': 4, 'score': 0.16653063893318176},
           {'label': 5, 'score': 0.017037797719240 19}]]
```

Conclusions

We achieved the best model for Experiment D when we integrated the Back Translation data augmentation technique with the fine-tuned model. The Data augmentation was very powerful in solving the imbalance problem that was obvious through the heatmap results in Experiment A and C. It improved the training F1-score, decreased the training and validation loss and resulted in a lower Mean Square Errors for the classes prediction. Finally, fine-tuning the AraBERTv0.2 model to fit our problem enhanced the results even more.

Future Directions

For future considerations, we will be:

- Appling Interpretability Analysis on our model to understand how it is making its predictions and which words have the most powerful effect on the model's decision.
- Experimenting with more hyper-parameters and see the combined effects of changing more than one hyperparameter together.
- Implementing a utility application for our model, most probably a website, that will be available for the public usage and will contribute to the Arabic Sentiment Analysis Research.

Contact Information

Iman Attia, Ahmed Essam
The American University in Cairo
Email: iman.elsadany@aucegypt.edu
Phone: +201115870440

References

1. J. Devlin and M.-W. Chang, "Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing," *Google Research*, 02-Nov-2018.
2. Al-Twairesh,N. The Evolution of Language Models Applied to Emotion Analysis of Arabic Tweets. Information 2021,12, 84.
3. Papers with code - hard dataset. HARD Dataset | Papers With Code. (n.d.). Retrieved February 22, 2023, from https://paperswithcode.com/dataset/hard
4. Papers with code - ASTD dataset. ASTD Dataset | Papers With Code. (n.d.). Retrieved February 22, 2023, from https://paperswithcode.com/dataset/astd
5. Papers with code - ArSentD-LEV dataset. ArSentD-LEV Dataset | Papers With Code. (n.d.). Retrieved February 22, 2023, from https://paperswithcode.com/dataset/arsentd-lev
6. Papers with code - LABR dataset. LABR Dataset | Papers With Code. (n.d.). Retrieved February 22, 2023, from https://paperswithcode.com/dataset/labr

Acknowledgements

We would like to thank Professor Mostafa Yousef for his support throughout the project. Also, we would like to acknowledge the efforts of the Teaching Assistant Sherif.