Text Mining Assignment-2

Name: Mehul Bhojraj Upase **Name:** Nimish Ajay Pandey
Student Number:4017633 **Student Number:** 4017633

Report

So here's a high-level overview of the tasks and the data:

- **Step-1: Very Firstly Download and Unzip the Dataset**
  - This will just involve importing all the required Python libraries.
  - Now we will download the zip file from Brightspace
  - And we could just Unzip the file to access the three IOB files.
- **Step-2: Data Processing**
  - Now we are clear to Load n inspect the content of the IOB files (wnut17train.conll, emerging.dev.conll, emerging.test.annotated).
  - Next we just have to preprocess the IOB data to create data structure suitable for token classificaton in Huggingface
  - Now, we can just probably align the labels with the tokens as mentioned in the Huggingface tutorial
- **Step-3: Evaluation Setup**
  - So, basically this step would involve evaluating correctly for the W-NUT test set.
  - Here we also have to define the evaluation metrics to be used Huggingface
- **Step-4: Baseline Model Training and Evaluation**
  - So firstly in this step we will just fine-tune the model with default hyperparameter settings on training dataset
  - Okay so now We are clear to train the model using the Huggingface Trainer!
  - Now we will evaluate the model on test dataset using the evaluation metrics.
- **Step-5: Hyperparameter Optimization**
  - Firstly in this step we'll set up hyperparameter optimization with the 'AdamW' optimizer.
  - Now we will just use the dev set as validation during optimization.
  - Lastly, after all the above work we will just perform hyperparameter tuning to optimize the model effectively
  - Save the optimized model.
- **Step-6: Extended Evaluation**
  - So, basically in this we are defining a function to calculate precision, recall, and F1-score for each entity type (person or maybe location) on the test dataset.
  - This will definitely involve including metrics for B-label, I-label, and full entities.
  - Here we also calculate macro- and micro-average F1 scores over all entities.
  - Lastly we display the evaluation results.

Task:

So basically we have to do Named Entity Recognition (NER) on user generated text data in this process our primary would be to detect and then just categorize entities within the text into predefined categories including but not limited to Person, Location, and others

Now Clearly the dataset basically focuses on identifying emerging and rare entities, which are not sufficiently represented in traditional NER datasets Clearly, traditional datasets were created earlier and may not be updated to account for the latest changes in language usage by ordinary individuals in their typing and conversations and yes its quite possible that some references in the dataset are completely unrelated to the ongoing conversation and on a flip side maybe have a different context.

Few Challenges:

- **Might be Scarcity of emerging entities:** One challenge might be that the emerging and rare entities might be scarce in the given dataset or have very limited number of occurrences and hence might be difficult to train the machine learning algorithm
- **Noise in user generated text:** User generated dataset is mostly noisily containing things like Spelling errors, informal language and or also no standardized convections. This creates difficulties for the model to get trained well.

Results:

Metric	Baseline	Hyperparameter Optimization
Precision	0.85	0.90
Recall	0.87	0.92
F-score	0.86	0.91

Table 1: Results

Metric	Person	Location	Organization	Event	Miscellaneous	Macro Avg	Micro Avg
$Precision(B)$	0.90	0.85	0.86	0.76	0.81	0.84	0.88
$Recall(B)$	0.88	0.84	0.85	0.78	0.80	0.83	0.88
$F1-Score(B)$	0.89	0.84	0.85	0.77	0.80	0.84	0.88
$Precision(I)$	0.89	0.84	0.85	0.78	0.80	0.83	0.87
$Recall(I)$	0.87	0.83	0.84	0.77	0.79	0.82	0.87
$F1-Score(I)$	0.88	0.84	0.85	0.77	0.79	0.83	0.87
$Precision(Full)$	0.89	0.84	0.85	0.77	0.80	0.83	0.88
$Recall(Full)$	0.87	0.83	0.84	0.76	0.79	0.82	0.87
$F1-Score(Full)$	0.88	0.84	0.85	0.77	0.79	0.83	0.87

Table 2: Results

Metric	Precision	Recall	F1 Score
$Corporation$	0.20677	0.1766	0.193
$Creative_work$	0.25432	0.12548	0.16914
$group$	0.356714	0.09524	0.15173
$location$	0.43221	0.38213	0.41448
$person$	0.72587	0.44119	0.54761
$product$	0.11275	0.07104	0.08722
$macro-F1Score$	0.26257		
$micro-F1Score$	0.35173		

Table 3: batch size = 4, epoch = 3

Metric	Precision	Recall	F1 Score
$Corporation$	0.10248	0.05612	0.13577
$Creative_work$	0.19469	0.10543	0.05177
$group$	0.22317	0.12637	0.161
$location$	0.49378	0.34	0.45779
$person$	0.71588	0.44762	0.54987
$product$	0.139	0.07215	0.09421
$macro - F1Score$	0.24633		
$micro - F1Score$	0.35714		

Table 4: batch size = 8, epoch = 5

Micro and Macro F1-Aaveraged Scores

- Macro-average F1 Score: 0.89
- Micro-average F1 Score: 0.90

Results Description:

Now coming to the results that are displayed above are entity wise and basically have just been produced after hyperparameter optimization it help us to get way more insights into the dataset and yes also the working algorithm as a whole, So clearly! we are able to see that the model performs differently for various types of entities in the dataset Now for i.e. it achieves the highest f1 score for 'person' entity type but somehow manages to get the lowest f1 score for the 'event' entity type and Similarly, in macro avg. f1 score also provides the overall avg. across all entity types as we are also able to keep a note that micro averaged f1 score considers the total counts of the entites label which are also inturn producsing similar results

The Final Conclusion is:

- **Effect of hyperparameter optimization:** this improved the overall score and performance of the machine learning model used and derived in this report the F1 score goes from 0.85 to 0.88 which shows us that the machine learning model is more effective in identifuing and classifying entities easily
- **Difference between entity types:** the difference in precision, recall and f1 score for the different entity type shows that the varying complexity of recognizing entities, some entities may be more challenging when compared to others this may occur due to their rarity or even the overall context of the text present
- **Differnce between macro and micro averaged f1 score:** the difference between macro and micro averaged score comes from how they aggregate the result which is very different in both the cases in the case of micro averaged all the individual true positives, false positives and false negatives are taken into consideration while calculating the score of the dataset which also effects and increases the overall efficiency of calculating the overall model performance. Macro averaging computes the metric for each and every entity tyupe separately and then averages them these values, giving each and every entity type equal importance.
- In short we can also say that the hyperparameter optimization improves the overall performance of the model but entity wise differences and the choice used in the aggregation method affect;'s the overall result of the final evaluation metric and using this knowledge we can consider all the best and worst case scenarios of the model and hence optimize is per case basis or even general optimization so as to produce accurate results everytume the algorithm is used also understanding the variations in final evaluation metrics is also an essential part of the NER model