Final Project : Contradictory, My Dear Watson

Department: Applied Statistics

Student ID: 2020122062

Name: Juwon Lee

Introduction

: Introduce the competition, its importance, and the problem it aims to solve. State the objectives of your project clearly.

In recent years, the field of Natural Language Processing(NLP) has seen steady growth. Among various NLP problems, Natural Language Inferencing(NLI) is a task that analyzes how a pair of sentences, consisting of a premise and a hypothesis, are related. If machines acquire the ability to analyze relationships between sentences, it could greatly aid in tasks such as fact-checking, detecting fake news, and analyzing text, among others. Also, NLI can contribute in several meaningful ways such as legal document and contract analysis, enhancement of chatbots and virtual assistants.

Two sentences can be connected in one of three ways: one may imply the other, one may oppose the other, or they may have no connection at all. Kaggle competiton ‘Contradictory, My Dear Watson’ aims to create an NLI model that assigns labels of 0, 1, or 2(corresponding to entailment, neutral, and contradiction) to pairs of premises and hypotheses. The train and test set include text in fifteen different languages. By carrying out the project in competition ‘Contradictory, My Dear Watson’, we can not only solve NLI problems but also lay the groundwork for artificial intelligence to contribute to various fields.

Methodology

: Describe your approach, including data preprocessing, model selection, and any challenges encountered. Provide rationales for your methodological choices.

As the first step, I conducted an Exploratory Data Analysis (EDA) on the given data. The training dataset consisted of a total of 12,120 sentence pairs. The columns in the training data were ‘id’, ‘premise’, ‘hypothesis’, ‘lang_abv’, ‘language’, and ‘label’. The distribution of labels in the training set was as follows: entailment with 4,176 pairs (~34%), contradiction with 4,064 pairs (~34%), and neutral with 3,880 pairs (~32%). The dataset covered 15 languages, with English accounting for 57% and the remaining languages (Chinese, Arabic, French, Swahili, Urdu, Vietnamese, Russian, Hindi, Greek, Thai, Spanish, Turkish, German, and Bulgarian) each contributing 3%.

Next, I proceeded with data preprocessing. The training data was split into training and validation sets with a ratio of 0.2, and a DatasetDict was created to structure the dataset into training, validation, and test sets. Initially, I used the google-bert/bert-base-multilingual-cased model. This model, developed by the Google BERT team, is a pretrained model for 104 languages with the largest Wikipedia datasets, using the Masked Language Modeling (MLM) objective. Using

the transformers library, I imported the AutoTokenizer, AutoModelForSequenceClassification, and DataCollatorWithPadding classes to configure the model, tokenizer, and padding options.

The tokenizer was then used to preprocess the data into a format that could be input into the model. Each pair of premises and hypotheses was concatenated using a 'SEP' token. Subsequently, I used the Trainer API to train the model. However, the model's accuracy plateaued at around 0.5, which was significantly lower than expected. This result led me to consider replacing the model.

Next, I opted for the FacebookAI/xlm-roberta-base model. Developed by Facebook AI, this model utilizes larger and more diverse datasets than the google-bert/bert-base-multilingual-cased model. Built on the RoBERTa architecture, it leverages enhanced training techniques like Dynamic Masking, making it more suitable for my project. After hyperparameter tuning, I achieved an accuracy of 0.7182. While this result placed me in the top 40% of the leaderboard, achieving the top 10% required an additional 0.18 improvement in accuracy. To meet this goal, I sought out a new model.

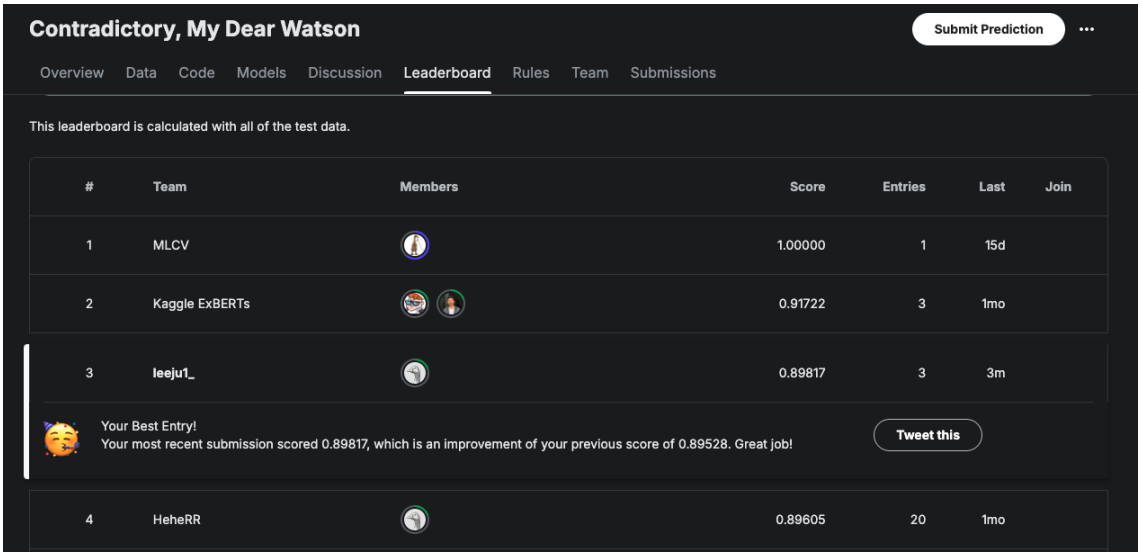
To further improve accuracy, I chose the MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7 model. This model incorporates the DeBERTa architecture, utilizing the Disentangled Attention Mechanism to separate content and positional information for finer contextual representation. It also employs the Enhanced Mask Encoder to improve Masked Language Modeling, making it particularly effective for NLI tasks. Applying this model initially yielded an accuracy of 0.85. To enhance accuracy further, I increased the train-test split ratio from 0.2 to 0.3, which resulted in an accuracy of 0.86025.

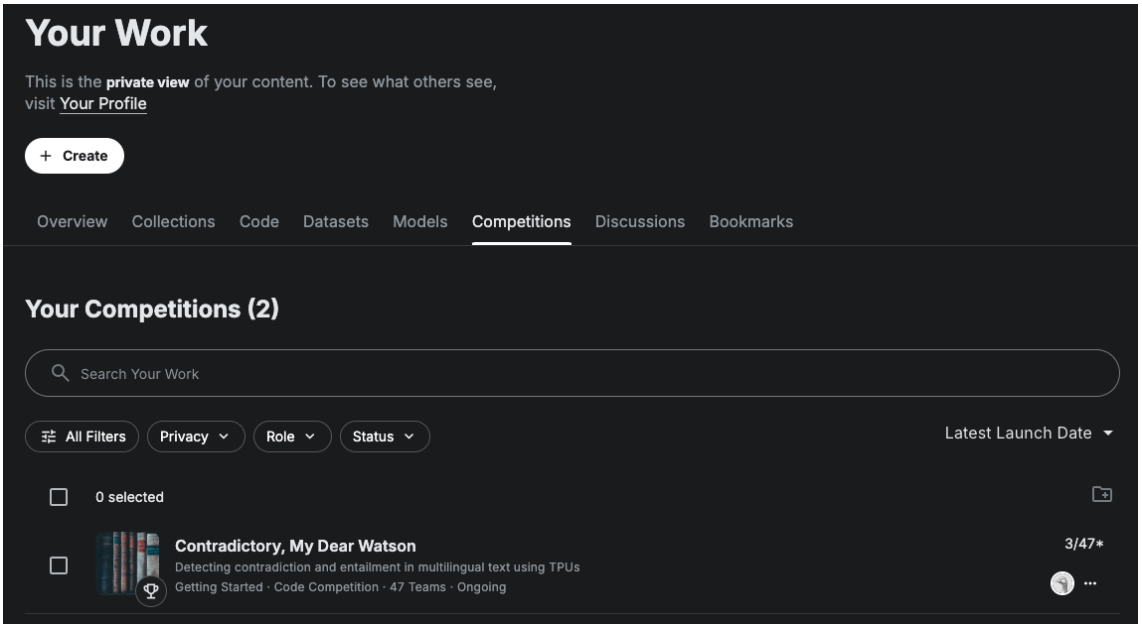
However, there was still a 0.03 gap to reach the top 10% accuracy on the leaderboard. To close this gap, I attempted to increase the batch size to 32. Unfortunately, this led to a CUDA Out of Memory error, as the GPU lacked sufficient memory to allocate resources for training. I implemented one of the fundamental methods for improving model performance: training data augmentation. I decided to load additional XNLI data and incorporate it into the existing training dataset. The Cross-lingual Natural Language Inference (XNLI) corpus is the extension of the Multi-Genre NLI (MultiNLI) corpus to 15 languages. Adding 10,000 sentence pairs to the training set also resulted in an Out of Memory error.

As a final approach, I tried training separate models focused on English and multilingual data respectively. Considering the dataset's characteristics, where English accounts for half and other languages for the other half, as well as methods adopted from previous projects, I thought this approach might be viable. For the English dataset, I used the sileod/deberta-v3-base-tasksource-nli model, while for the multilingual dataset, I continued using the MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7 model. At last, I achieved an accuracy of 0.89817, which led me to rank #3 out of 47 participants.

Results

: Present your results and compare them to baseline performances, using appropriate metrics. Screenshot of your rank should be included here.





The baseline performance for the "Contradictory, My Dear Watson" competition can be found in the Tutorial Notebook(https://www.kaggle.com/code/anasofiauzsoy/tutorial-notebook). The baseline model used was google-bert/bert-base-multilingual-cased, achieving a best score (accuracy) of 0.63349. The competition set accuracy as the evaluation metric to provide a more intuitive understanding of the model's performance.

My results(Score: 0.89817) significantly surpassed the baseline model in terms of accuracy. Additionally, I ranked in the top 10% (3th out of 47) among all participants in the competition.

Discussion

: Interpret your findings, discuss the significance of your results, and reflect on the project's limitations and what you learned.

The final model developed in this project demonstrates the effectiveness of using language-specific models for Natural Language Inference (NLI) tasks across different languages. By fine-tuning separate models for English and multilingual data, this approach addresses the unique

linguistic characteristics of each language, ensuring better performance on language-specific nuances.

English-centric models, like those trained on SNLI and MNLI, excel in English, where most NLI resources and pretrained models are focused. In contrast, multilingual models like mDeBERTa are better suited for handling datasets with multiple languages, such as XNLI. This dual-model approach allows for specialized performance on English while maintaining broader applicability across other languages.

However, the project faced some limitations. A single model capable of handling both English and multilingual datasets was not feasible due to architectural differences. Additionally, restricted GPU memory in the Kaggle notebook environment prevented the use of data augmentation techniques, limiting model improvement. Further, external datasets like XNLI and SNLI were not fully leveraged, which could have enhanced model performance and generalization across languages.

Key lessons from the project include the importance of adapting models to linguistic properties and the trade-off between specialization and generalization. Resource limitations significantly impacted experimentation, highlighting the need for more computational power. Combining both English-centric and multilingual models offers a balanced solution for multilingual NLI tasks.

Conclusion

: Summarize your insights, discuss the broader implications of your work, and suggest avenues for future research.

By doing this project, I was able to explore the challenges and opportunities of applying Natural Language Inference (NLI) across different languages. By utilizing language-specific models for English and a multilingual model for other languages, we developed a framework that effectively addresses NLI tasks with a focus on linguistic diversity. The results demonstrate the potential of specialized models to capture the unique complexities of different languages while maintaining strong performance across various linguistic contexts.

The key takeaway from this project is the importance of adapting models to the linguistic properties of the data. While English-centric models, such as those fine-tuned on SNLI and MNLI datasets, perform well in English, they may not be as effective for multilingual tasks. On the other hand, multilingual models like mDeBERTa, which can handle multiple languages simultaneously, offer a more generalized solution, but they may not always achieve the same level of performance as specialized models in specific languages. This contrast highlights the need for a hybrid approach in NLI tasks that combines the strengths of both types of models.

Beyond technical performance, this project emphasizes the importance of resource availability in training and deployment. The limited GPU memory in the Kaggle environment prevented us from experimenting with data augmentation techniques, which could have further improved model robustness, especially in low-resource languages. Additionally, the lack of external datasets like XNLI or SNLI for multilingual languages restricted the model's ability to capture a broader range of linguistic features and relationships. These limitations provide valuable insights into the constraints faced when working with large-scale NLI tasks and the need for additional computational resources to address them.

Looking forward, there are several promising directions for future research. One possibility is integrating both English-centric and multilingual models into a unified system that could dynamically select the appropriate model based on the language of the input data. This would

enhance efficiency and ease the management of multiple models. Expanding the dataset to include more diverse languages and augmenting the data could further improve model performance and robustness, particularly in underrepresented languages. Another potential direction is experimenting with advanced models that incorporate few-shot learning to reduce reliance on large-scale annotated datasets.

Ultimately, this project reinforces the idea that while multilingual models are a powerful tool for handling multiple languages, there is still significant value in using specialized models for specific tasks. By carefully selecting the right model for the appropriate context, we can achieve better performance and more effective solutions in Natural Language Processing, especially in complex tasks like NLI.

References

: Include all references in APA format.

1. google-bert. (n.d.). BERT multilingual base model (cased). Hugging Face. https://huggingface.co/google-bert/bert-base-multilingual-cased
2. FacebookAI. (n.d.). XLM-RoBERTa (base-sized model). Hugging Face. https://huggingface.co/FacebookAI/xlm-roberta-base#xlm-roberta-base-sized-model
3. MoritzLaurer. (n.d.). mDeBERTa-v3-base-xnli-multilingual-nli-2mil7. Hugging Face. https://huggingface.co/MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7
4. sileod. (n.d.). DeBERTa-v3-base-tasksource-nli. Hugging Face. https://huggingface.co/sileod/deberta-v3-base-tasksource-nli
5. ANA SOFIA UZSOY. (n.d.). Tutorial Notebook. Kaggle. https://www.kaggle.com/code/anasofiauzsoy/tutorial-notebook/notebook
6. YASHPURI_19. (n.d.). My_Dear_Watson. Kaggle. https://www.kaggle.com/code/yashpuri1912/my-dear-watson