HSLU Lucerne University of Applied Sciences and Arts

Course Project

NLP
Andreas Marfurt

Information Technology
25.11.2022

FH Zentralschweiz

Goals and Work Plan

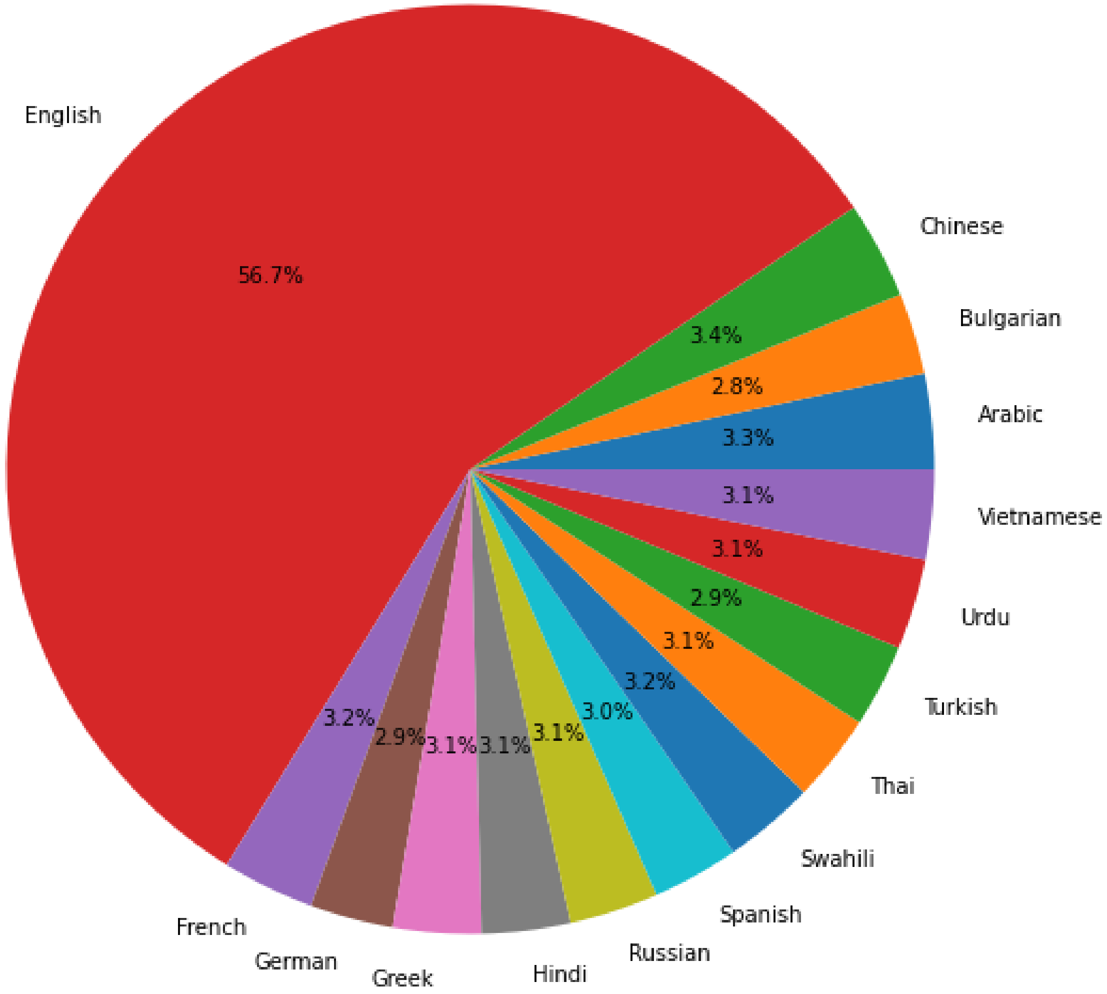
- Analyze the data
- Find relevant prior work
  - Potentially including implementations that can serve as baselines
- Determine appropriate models to try
- Start with some baselines
  - Compare with a non-neural network solution
  - Finetuned model from HuggingFace
- Train your own model(s)
  - Train(/finetune) at least 1 neural network yourself
- Evaluate and compare models
- Write up the results so that readers can understand and reproduce them

Data

Kaggle dataset: Contradictory, My Dear Watson

- Multi-lingual natural language inference
  - Labels: 0 = entailment, 1 = neutral, 2 = contradiction
  - Features: id, premise, hypothesis, language abbreviation, full language name, label
- Original training data: 12120 examples
  - I divided this into 10908 training (train.csv) and 1212 validation (valid.csv) examples (90/10 split, see Project documents)
- test.csv: 5195 examples that you have to classify
- The example notebook uses the `bert-base-multilingual-cased` model and shows how to work with the data (unfortunately in TensorFlow)
  - `bert-base-multilingual-cased` is a good baseline for you as well!
  - The example uses TPUs, but we will use our laptops and Google colab (free GPUs)

Language Distribution



Train/Validation/Test Splits

- Train: Use this data to train your models
- Validation: Here you can tune and compare different versions of your model
  - This data is never used to train your model
- Test: Once you have selected a model and its hyperparameters, test it on the test data for the final comparison
  - Tuning hyperparameters on the test set is considered cheating (in industry/research) and gives you a false picture of how your model will perform on truly unseen data

Task: Natural Language Inference

- Predict whether the premise entails/is neutral towards/contradicts the hypothesis
- Do an error analysis
  - What types of errors does your model make?
    - Manually inspect a few examples where your model makes errors
    - Group errors (e.g. by true label, predicted label, language). For which groups/combinations does it make the most mistakes?
  - Speculate on why your model makes those errors
    - It is very difficult to determine with certainty why a neural network makes certain errors. This is out-of-scope for this project. But can you detect certain patterns in your error analysis?

Including Prior Work

- It is ok to use prior work as baselines, but it needs to be cited!
  - This includes HuggingFace pretrained/finetuned models
- Train your own model(s): At least 1 neural network
  - Clearly separate your own work from work done by others

Deliverables and Deadline

- No intermediate presentation
- Project presentation: 20 minutes + 10 minutes for questions
  - Date: December 23, 2022 (last day of semester)
  - Every group member presents something
- Report: Fill out the canvas
  - Hand in: December 23, 2022, 1 PM
  - Page limit: 20 (keep font & font size of canvas)
    - Includes graphs, diagrams, …
- Code
  - Hand in together with report
  - Jupyter notebook(s) including documentation (as in exercises)
  - Model checkpoint of your trained model

Grading

- Report: 50%
  - Structure given by the canvas
  - Motivation for selection of algorithm
  - Presentation of results
  - Error analysis
  - Grammaticality and formatting, adherence to length limit
- Presentation: 40%
  - Overview of project: task description, data analysis, selected method, results, analysis
  - Lessons learned: what went well? What was difficult? Was something surprising?
  - Clear presentation, clean slides
- Code: 10%
  - Good documentation
  - Reproducibility of results
  - Clean coding style
  - Efficiency
  - Correctness
- Bonus: 10%
  - Submit your predictions to the Kaggle leaderboard and see how your model compares to others
  - Include a screenshot in your presentation and/or report
  - Bonus points are computed from the leaderboard position relative to Kaggle and HSLU-NLP models