Natural Language Processing

Topic Overview

Sequence classification/labelling is one of the most in-demand functions that has been implemented for solving a wide variety of problems. It is the task of predicting a class label given a sequence of observations. In many applications such as healthcare monitoring or intrusion detection, early classification is crucial to prompt intervention. The most commonly known scenario of sequence classification in natural language processing (NLP) is labelling text with named entities like PERSON, LOCATION, ORGANIZATION. Sequence classification can be used to model many problems in information extraction and is applicable to e-commerce, dialogue assistants, error recognition in machine translation (word-level quality estimation), and so on.

you aim is to perform sequence classification for abbreviation and long form detection, where ABBREVIATIONS are labelled by AC and LONG FORMS are labelled by LF. Since multiple tokens/words can belong to the same long form, this problem has adapted the labelling schema known as the BIO format[1].

In our data, tokens labelled with B-O (or 'O') indicate other tokens which are neither abbreviations nor long forms. B-AC signifies that token is an abbreviation/acronym, while B-LF signifies that a long form 'begins' with this token. I-LF label signifies that the token is 'inside' of a long form. The data also contains the part-of-speech (POS) tag which are optional to use. An example segment from the data (syntax: <token><space><POS><space><BIO tag>) may look like:

EPI PROPN B-AC
= PUNCT B-O Echo
NOUN B-LF planar
NOUN I-LF imaging NOUN
I-LF
. PUNCT B-O

Sequence classification is useful for use cases where one needs to extract information from a set of documents given labelled data for training and/or validation. The current dataset is sourced from scientific literature in the PLOS journal articles and belongs to the biomedical domain.

During the prediction process (*i.e.,* inference) the input tokens can be assigned with only one label.

---

[1] Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition - ACL Anthology

Your task is to build a sequence classifier prototype for this data provided to you, and in this instance is the PLOD dataset consisting of 50k labelled tokens.

The details of the dataset, including how to download, are given below:

- surrey-nlp/PLOD-CW · Datasets at Hugging Face
- Labels: B-O, B-AC, B-LF, I-LF
- Maximum token sequence length in the dataset: 323

Optional Dataset: surrey-nlp/PLOD-filtered · Datasets at Hugging Face

Please NOTE: The original PLOD dataset is significantly large, and PLOD-CW is a further filtered version of PLOD-Filtered. For experimentation, you can choose to take more training instances from PLOD Filtered but make sure you do not duplicate them.

Experimentation

You will research and experiment on different ways, to prepare data and train the model. All experiments and documentation must be done in a Jupyter notebook. Having multiple Jupyter notebooks for different experiments is fine, but I recommend a single notebook which clearly delineates each experiment with headers and sub-headers within for further description, and is able to call other notebooks as functions/modules.

It is expected to produce in sections the following. Note that it is important that your PDF report clearly shows a heading for each of these numbered sections, with the heading number and section title. Also, start each section on a new page Conciseness is appreciated in the PDF report: do not include long dumps of log files or code.

1. Analyse and visualise the dataset – produce charts and document observations. Use papers in the reference to think of how you can analyse the dataset.

2. Experimentation with four different experimental setups, where you might be trying out different options such as (listing more than four different experiments here, so choose four). Your PDF report should have a subheading for each of the 4 experiments, numbered 2.1, 2.2, 2.3, and 2.4.
    I. data pre-processing techniques – tokenise (e.g., will you use n-grams?). In case of pre-trained language models, you will be using their own tokenizers.
    II. NLP algorithms/techniques – explain your choice (e.g., I am comparing SVM, HMM, CRF, RNN, Fine-tuning (FFNN), *etc.* to understand their advantages and disadvantages when trained with the dataset…). Experiment with at least two algorithms.

III. text encoding/transformation into numerical vectors – justify choices (like tf-idf, word2vec, glove, fasttext or a pre-trained language model, and/or other relevant encoding methods). Experiment with at least two methods.

IV. Any additional train/validate dataset from the optional dataset – how much did you extract and why (try different proportions and observe the difference in the evaluation and how much you can fit on the free tier GPUs). This is completely optional but if done well, this should enrich your final analysis and report.

V. choices of loss functions and optimisers – explain your choices with facts from the results.

VI. hyperparameter optimisation – what are the most appropriate values (e.g., learning rate, training cycles, etc., depending on the algorithm)

VII. finetuning vs full training – which one is more appropriate (it might depend on the dataset)

VIII. other setups that you might find relevant – make sure to justify why you chose this experiment variation.

NOTE: You will submit 4 full experiments consisting of everything from data pre-processing to evaluation using the F1-metric. Each experiment should have two/three comparisons.

3. Analyse testing for each of the four experiment variations conducted above (this refers to accuracy testing, not software testing) – show visuals, such as confusion matrix or other relevant metrics for each experiment. Use F1-score as primary evaluation metric. Perform an error analysis on the predictions obtained. Your PDF report should have a subheading for each of the 4 experiment variations, numbered 3.1, 3.2, 3.3, and 3.4.
4. Discuss best results from the testing, on all the experiments you conducted and mention if there was any need to adjust any variables and re-run the experiment
5. Evaluate the overall attempt and outcome – this goes beyond the accuracy of the models, so some important questions to consider here are:
    a. "Can the models you built fulfil their purpose?"
    b. "What is good enough F1/accuracy?"
    c. If any of the models did not perform well, what is needed to improve?
    d. If any of the models performed really well, could/would you make it more efficient and sacrifice some quality?

    Make sure you justify the choice between the most accurate against the most effective solution

Important to understand

For example, if you choose to potentially perform these experiments:
Experiment1: Comparing features/vectorization methods
System 1: No preprocessing, pre-trained language model, Finetuning with FFNN vs.
System 2: No preprocessing, word2vec-based feature extraction, FFNN, evaluation… vs.
System3: No preprocessing, GloVe-based feature extraction, FFNN, evaluation.
(This is one experiment with three different vectorization methods and other variables like algorithm and pre-processing are the same).
Experiment2: Comparing algorithms

All systems use same vectorization method, but the algorithms can be SVM vs FFNN vs RNN.
Experiment3: Comparing loss functions/optimizers
Use same pre-trained language model while fine-tuning but change loss functions and optimizers. Find some novel loss functions you can perhaps implement and see its rewards in performance.
Experminet4: Additional data instances with different best performing systems from above Use additional training and validation data to try and improve performance. Improvement may not be guaranteed but you must document results and analyse.

The marks for Q2 will be rewarded based on your experimental choices and methodology while for Q3 they will be rewarded based on the testing you perform for these experiments and how nuanced or detailed it is. Hope this makes things clear.

Since some needed lectures won't be taught until late in the year, it is expected that you will still progressively and continuously work on the coursework. Each week there will be lab exercises that can help you with different parts of the coursework (e.g. data preparation, visualisation, data transformations, featurisation and other will be taught from week 2). So, it is NOT recommended to wait until all the lectures are taught before you get started.

Deliverables

You will need to submit a Jupyter notebook documented appropriately, but this needs to be submitted in two formats:

1. ".ipynb" notebook file (plus any helper files you might use) [mandatory File 1, inside ZIP]
2. pdf report (your report on experiments) with each experiment described in detail. It should contain either the results in tables or screenshots of results from notebook. It must have your error analysis of mispredictions made by the best model, and any other relevant sections based on [mandatory File 2, outside ZIP]
3. DO NOT print the entire dataset or any list/dictionary etc on the notebook. Instead, just print the top few (maybe 10?) records.
4. DO NOT submit the dataset(s), just add reference(s)
5. DO NOT submit the trained model(s)

The notebook should contain visuals (where appropriate) to support tasks such as: label data distribution, histogram comparisons, text samples, classification accuracy curve, confusion matrix, etc. If there are library dependencies, please also include a requirements file.

References:

Comparison of named entity recognition methodologies in biomedical documents | BioMedical Engineering OnLine | Full Text (biomedcentral.com)
HiNER: A large Hindi Named Entity Recognition Dataset - ACL Anthology
PLOD: An Abbreviation Detection Dataset for Scientific Documents - ACL Anthology Token classification - Hugging Face NLP Course

Report:

Question1: Extensive analysis of the dataset(s) containing clear visuals, observations, and explanations.

Question2 (for each experiment): Great approach to experimentation. The methodology is well justified and well documented, and the implementation runs the experiments successfully without any errors.

Question 3 (for each experiment): Comprehensive testing and good explanation of the test results with supporting visuals, which indicates if a model/technique is adequate.

Question 4: Clear explanation of the outcomes and meaningful guidance on ways to make improvements.

Question 5: Well articulated evaluation of the overall attempt, and great explanation on the choice between the most accurate against the most effective solution.