Natural Language Processing
COM3029 & COMM061

Coursework

2023-2024

Table of Contents

Please note: The coursework is subject to minor changes depending on the suggestions from checker/external examiner.

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.

In this coursework, 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.

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.

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

Structure

The module is assessed 100% on this coursework. The coursework contains both group and individual contributions and is divided into three parts (with separate submission for the individual and group parts):

1. A group formation and plan of work (not assessed)
2. An experimentation part where each individual works independently, based on the group plan.
3. A final part where a basic prototype system is put together by the group, based on the results of the individual contributions.

You have been grouped on SurreyLearn, and must be aware of the members. It is highly recommended you start discussions as early as possible and come up with a concrete plan.

HINT 1: The methodology is much more important than the accuracy of the model. So, make sure you appropriately follow the taught methods for performing the experiments rather than spending too much time improving the accuracy of the model.

HINT 2: Work continuously as a group, but make sure the individual part is attempted alone and submitted separately. The team should be there to support any of the members that has issues, and in some cases the original plan can change if needed.

HINT 3: The deadlines are there for submitting that different required parts, but this should not stop you from moving forward to working in the next part! For example, if everyone has finished working on the experiments on week 8, then there is no need to wait till week 10 to build the prototype required for the final part. Some parts of the group work could start even sooner.

HINT 4: Although you are allowed (and I even encourage you) to search the Web for finding examples, researching solutions, and using existing material such as code, data, definitions, diagrams, lecture slides, lecture labs, and other that will be used directly or indirectly in your submissions, all such material must be referenced in a clear and concise way.

No additional infrastructure should be requested for coursework experiments. You are to use GPUs provided in the Heron lab or free tier GPUs.

None of the material in your submission, viz., code, written text, comments for code, and so on, should be generated via language model(s). This will be considered a serious case of academic misconduct and is liable to be reported to the University immediately.

Group - Declaration

Weight: 0% of the marks
Submission date: Week 5 (Friday, 15th March, 11 PM)

You will start this project as a group, then work individually, and then eventually get back together to complete the last part. This first part is important as it will set the group and work structure for the next two parts.

Discuss as a group / decide on the following:

1. Discuss and gain a collective understanding of the sequence labelling (or token classification) task. Ensure you understand the BIO schema of labelling and can understand the segments and tags in PLOD-CW dataset provided at the link above.

2. Look at number of instances in each dataset split, *viz.,* train, validation, and test.

3. Discuss if you will use any additional number of training instances as a group and extract those instances from the optional dataset (PLOD-Filtered).

4. Plan the experiments, while keeping in mind your resources and time constrains, and explain the scope and priorities.

5. Decide on a common development environment (code repository, python libraries needed, etc.) and list the choices made.

6. Divide individual tasks and explain how you are planning to work together. Each group member could potentially focus on a different experiment setup (*i.e.,* this is part of the individual assessment). This might be experimenting with different:
    a. data set preparations (pre-processing and/or featurisation)
    b. algorithms
    c. pre-trained models (transfer learning)
    d. setup of the hyperparameters
    e. any other such relevant experimental variations (if needed)

Deliverables

Each group will need to submit a one-page document (min size 10 font) including statistics of additional instances extracted, if any, plan for the experiments, the development environment, and the individual tasks per group member. Any front cover or appendix can be added if relevant. This needs to be submitted by ONE of the group members on SurreyLearn, before the set deadline. The one-page document must be named as group_<number>.pdf, *e.g.,* group_01.pdf

Individual - Experimentation

Weight for COMM061 students: 50% of the marks
Weight for COM3029 students: 70% of the marks

Submission date: Week 9 (26th April, Friday, 4 PM)

Each student will individually research and experiment on different ways (or even the same way if the group feels is appropriate), to prepare data and train the model. The choice of the different individual experiments (tasks) should be discussed and decided as a group, as defined in the group declaration report, but attempted and documented separately by each individual. Group discussions and coordination should still continue for the group submission. If the experimentation plan changes (needs to be a group decision), then you will need to just provide some justification in your group submission report. So, it is ok for the group to change the original plan without incurring any penalties. 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.

For the individual experiments, each student is expected to produce in sections:

1. Analyse and visualise the dataset – produce charts and document observations (5 marks). 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):
    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.

Since this will be a subpart of a group's experiment – the implementation, methodology and critical thinking is what matters here, and not if you (individually) covered all possible experimental setups (24 marks total, 6 marks per experimental setup).

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. (12 marks, 3 marks per variation)

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 (4 marks, 1 mark per variation)

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 (5 marks)

Important to understand

For example, if you choose to potentially perform these experiments:
Exp1: Comparing features/vectorization methods
System 1: No preprocessing, pre-trained language model BERT, Finetuning (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).
Exp2: Comparing algorithms
All systems use same vectorization method, but you changed the algorithms like SVM vs FFNN vs RNN.
Exp3: 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.
Exp4: 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 experiment and experimental methodology while for Q3 they will be rewarded based on the testing you perform for these experiments and how nuanced/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 till 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. Additional notebooks or Python files can also be included, but make sure you zip the files together before submitting. If there are library dependencies, please also include a requirements file. Only zip the code related file together, DO NOT put the pdf report in the zip file, so submit this separately, so that it can be checked for plagiarism. This should be submitted by each student independently on SurreyLearn; before the deadline.

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

Group - Deployment

Weight for COMM061 students: 50% of the marks
Weight for COM3029 students: 30% of the marks

Submission date: Week 13 (24th May, Friday, 4 PM)

This is the last part of the coursework assessment, where students get back together and combine their individual findings, choose and deploy the best solution, and build a pipeline that will train, deploy and monitor the model(s). All this work needs to be demonstrated and documented in a Jupyter notebook, together with any additional Python files that you might need to build. Having multiple Jupyter notebooks for different tasks is also fine. Discuss your best performing model and approach ONLY after you have submitted the individual coursework, to be able to choose the best model as a group. Some of these tasks can be started before submitting individual coursework.

Tasks:

1. Research different model serving option(s) and explain what the right choice for your case would be and why. Your research choices and exploration must be visible in the report submitted. (5 marks)

2. Build a web service (based on your previous choice) to host your model as an endpoint and make sure to explain the architectural choices (running the service locally on your machine is also sufficient). (10 marks)

3. Build some functionality in a notebook to perform testing on the deployed endpoint (i.e., some client function to consume the service via HTTP) and document your process and findings (in the notebook). No need for any UI here, so just command line interaction will suffice. (10 marks)

4. Discuss the performance of the service you implemented (i.e., perform some stretch testing to identify its limits), justify the good and bad points you discovered, and make recommendations for possible ways to improve the architecture. (5 marks)

5. Build some basic monitoring capability to capture user inputs and the model predictions, and store the inputs, model predictions and time/date of the interaction in a text log file (in a way that it can be parsed programmatically). There is no need to build additional functionality that will detect any concept drift, etc. (5 marks)

6. Build a basic CI/CD pipeline that will build and deploy the model when data or code changes. There is no need to trigger this automatically. If you can show in notebook/report how a manual execution script can be prepared, it will be sufficient. (5 marks)

7. 10 min screen recording (with voice description) of a demonstration of the group solution. The demonstration should show how you worked through the above tasks 1-6 (briefly) and it should clearly demonstrate the execution of the code. All members of the group should present a part of the demonstration (ideally the part they worked on) and this will be compulsory for everyone. (compulsory - 10 marks)

If for any reason none of the team members managed to complete the individual part of the assessment and you have no model to deploy, then you can either contact the lecturer to get a pre-built model, or alternatively you can use the pre-built model you found online. In either way, this should be documented appropriately in the notebook.

NOTE: The presentation recording is compulsory and if a member does not contribute, he/she will not be awarded any marks for the group submission. Please note this- Your contributions to the group MUST be clearly outlined in the video.

Deliverable

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 repot 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. Presentation- screen recording of group's video [mandatory File 3, outside ZIP]
4. DO NOT submit the dataset(s), just add reference(s)
5. DO NOT submit the model(s), just mention which model(s) was used and describe why it was used.

Additional notebooks or Python or other files can be included, but make sure you zip the files together before submitting. If there are library dependencies, please also include a requirements file. Only zip the code related files together, DO NOT put the pdf report in the zip file, so submit this separately, so that it can be checked for plagiarism. The presentation recording file should be submitted in a popular format such as mp4, avi, mov, etc. This needs to be submitted by one of the group members on SurreyLearn, before the set deadline.

Rubric

CW-1	Marks	Criteria
Q1	5	Extensive analysis of the dataset(s) containing clear visuals, observations, and explanations.
	3-4	Good analysis of the dataset(s), containing useful visuals and explanations enough to provide confidence on what is expected.
	1-2	Some analysis has been done, including visuals, but additions work was needed to ensure the dataset is fit for purpose.
	0	No analysis present, incorrect visuals and graphs.

Q2 (x4)	6	Great approach to experimentation. The methodology is well justified and well documented, and the implementation runs the experiments successfully without any errors.
	4-5	Good choice of a setup and good implementation that indicates how the variables need to be set. Good documentation but might need some more depth in parts.
	2-3	The setup is valid, but the variation chosen is still not representative to fully understand the correct setting. Documentation is not clear.
	0-1	The experimental setup is not clear or is redundant or does not contribute to the experiment.

Q3 (x4)	3	Comprehensive testing and good explanation of the test results with supporting visuals, which indicates if a model/technique is adequate.
	1-2	Some testing was done, but it does not really offer a certain picture of the model performance. The visuals used were limited.
	0	Not much testing was conducted, and the explanation was not appropriate. There were no correct visuals such as graphs to show results, or incorrect evaluation.

Q4	4	Clear explanation of the outcomes and meaningful guidance on ways to make improvements.
	2-3	Some explanation of the results achieved is given but needed mode details on how to improve performance.
	0-1	None or limited documentation of the results achieved. No explanation if there is further work needed to improve or not the models, or incorrect/incoherent explanations provided.

Q5	5	Well articulated evaluation of the overall attempt, and great explanation on the choice between the most accurate against the most effective solution.
	3-4	Good explanation on how the original problem is solvable or not and how the experiments helped to understand this.
	1-2	Some evaluation has been done, but not clear explanation on how or if the original problem is solved.
	0	Not much explanation on whether the original problem is solved or not. No evaluation of the overall attempt either.

CW-2	Marks	Criteria
Q1	5	Great research showing the different options, that was completed by explaining the right choice.
	2-3	Some research has been done, but the choices made are not well supported.
	0-1	Limited research done and no clear explanation on what is recommended.

Q2	9-10	Great architecture choice and implementation of the service. The model/mechanism can deploy easily and serve their purpose efficiently.
	7-8	Good architecture and decent implementation of the core mechanism but needed some additional functionality to allow for easier deployment.
	4-6	Ok implementation of the service, although not the most efficient. Some parts contained minor errors.
	0-3	There was not much implementation or there were many errors, and the model/mechanism would not deploy.

Q3	9-10	Thorough testing that covers multiple scenarios and ensures the correct operation of the service.
	7-8	Good testing overall but needed more clarification on the choices and interpretation.
	4-6	Some testing has been done but is not as relevant or it has some functional issues. The documentation is unclear in parts.
	0-3	The testing is non-existent or not working well. Not much documentation done either.

Q4	4-5	Great explanation of the key performance indicators when handling a variety of requests (size, speed, scalability, etc.) and good awareness of some good/bad points on the implementation.
	2-3	Some points were raised but some important points were missed. More was needed to explain the key performance indicators.
	0-1	Not much explanation of the key performance indicators, or justification of the good/bad points.

Q5	4-5	Solid functionality that captures the user inputs and predictions in a format that can be systematically used.
	2-3	Some functionality was working, but it was not capturing the data in a usable way.
	0-1	Not much functionality was implemented or working to collect the user input and the predictions.

Q6	4-5	Solid implementation of a CI/CD pipeline that builds and deploys models when the code or data changes.
	2-3	Some functionality is implemented but not fully covering the needs of a CI/CD pipeline.
	0-1	Not much functionality was implemented or working to build and deploy the model on data or code changes.

Q7	9-10	Great presentation that thoroughly demonstrated the implementation of the demo and how the choices were made.
	7-8	The recording was clear and concise but could have had some more context to explain some of the reasoning in the choices made.
	4-6	The recording was ok but was lacking clarity in different parts or needed explanation.
	0-3	The recording was non-existent or was of poor quality that could not be followed.