Natural Language Processing
COM3029 & COMM061

Coursework

Table of Contents

Topic Overview

Text classification is one of the most in-demand functions that has been implemented for solving a wide variety of problems. The most common scenario of text classification is for categorising documents in topics such as news items related to business, sports, politics or other events; another common scenario is for identifying spam in emails, or even understanding the sentiment on messages such as Twitter or when rating products on websites such as Amazon. During the prediction process (i.e. inference) the input text could eventually either be assigned with only one label (multi-class) or with multiple labels (multi-label), depending on how we need our application to behave.

Your task is to build a multi-class classifier prototype for a data topic provided to you, and in this instance is the GoEmotions dataset of 58k carefully curated comments extracted from Reddit, with human annotations into 27 emotion categories or "neutral". Although this dataset contains 28 labels (including the neutral label), you will be required to select part of the labels only (i.e. merging the remaining labels together in a new category).

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

- https://huggingface.co/datasets/go_emotions/viewer/simplified/train
- Number of examples: 58,009.
- Number of labels: 27 + Neutral.
- Maximum sequence length in training and evaluation datasets: 30.
- The emotion categories are: *admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise.*

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.

The group can be a minimum of 3 students and a maximum of 5. You will need to declare your group on SurreyLearn. It is recommended that you find your group as early as possible and talk about the coursework requirements. If you cannot find a group, you must tell the lecturer immediately so that you can be allocate to one.

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.

Group - Declaration

Weight: 0% of the marks
Submission date: Wednesday week 5

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 and decide on the following:

1. Choose your 13 labels (plus the neutral) from the dataset and discuss how to handle the remaining labels.

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

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

4. 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

You will need to submit a one page document (min size 10 font) stating the chosen labels, the 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.

Individual - Experimentation

Weight: 50% of the marks
Submission date: Wednesday week 9 (25th April, 23.59 hrs)

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. If the experimentation plan changes (needs to be a group decision), then you will need to just provide some justification. 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. Multiper Jupyter notebooks can be used but the code should be submitted in a way that only one notebook needs to be executed to replicate all experiments and observe results.

For the individual experiments, each student is expected to:

1. Analyse and visualise the dataset – produce charts and document observations (6 marks)

2. Experimentation with four different variations, 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?), normalise text, apply stopwords, and so on
   II. NLP algorithms/techniques – explain your choice (e.g., I am comparing SVM, FNN, etc. to understand their advantages and disadvantages when trained with the dataset…)
   III. text featurisation/transformation into numerical vectors – justify choices (like one hot encoding, and/or other relevant methods)
   IV. train/test/validate dataset splitting – how did you split and why (try different proportions and observe the difference in the evaluation)
   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.

   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 experiment).

3. Perform 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 (8 marks, 2 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 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 (8 marks)

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)
2. PDF report in a way to show executed outputs. Please see below for a template structure of this PDF.
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 model(s)
6. The notebook MUST NOT contain any output which DOES NOT help us evaluate your submission. Clear most of the outputs except visualizations, results, and perhaps a few lines from the output which show the commencement of training.

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 files 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.

PDF Report Structure

The PDF is going to be evaluated primarily. We will execute code too, but our primary focus shall be this final report. You can use screenshots/snippets from the notebook. The proposed structure is:

1. Title page (Name, URN, Team number etc.)
2. Group Declaration Submitted earlier, yes, even for this individual CW submission. It will help understand your label merging/mapping.
3. Data Analysis & Visualization
4. Experiments
    a. Experimental variation 1
        i. It should include clearly demarcated details of algorithms and hyperparameters used.
        ii. It should include the results on test set obtained in terms of classification report.

        iii. It should have confusion matrices.
        iv. Finally, your conclusion on results explaining which classes were misclassified the most, and which were misclassified the least. Treat this section as a discussion of results.
    b. Experimental variation 2 (all details same as above)
    c. Experimental variation 3 (all details same as above)
    d. Experimental variation 4 (all details same as above)
    e. Any more experiments done.
5. Best results from testing to be discussed here along with best algorithm, hyperparameters etc. (refer to point 4 from page 4 of this document)
6. Evaluate the overall attempt and outcome. (refer to point 5 from page 4 of this document)

Apart from this any additional analysis, error analysis would be appreciated too. Please do not make this PDF too long.

Group - Deployment

Weight: 50% of the marks
Submission date: Wednesday week 13 (24th May, 23.59 hrs)

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 but the code execution should be done from a single notebook.

Tasks:

1. Research different model serving option(s) and explain what the right choice for your case would be and why. (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 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, so a manual execution script 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. You can choose to record this together or merge video files for submission.

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)
2. pdf report in a way to show executed outputs.
3. DO NOT submit the dataset(s), just add reference(s)
4. DO NOT submit the model(s), just mention which model(s) was used

PDF Report Structure

1. Title page (Name, URN, Team number etc.)
2. Group Declaration Submitted earlier. It will help understand your label merging/mapping.
3. Details of best method from all individual experiments (algorithm, hyperparameters, final result)
4. Model serving options researched and final choice.
5. Performance discussion (point 4 on page 7)

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.

If any part of your code is generated via language models (*GPT*, Co-pilot, BARD, LlaMa, etc.) it MUST BE explicitly documented in comments above notebook cell.

Rubric

CW-1	Marks	Criteria
Q1	6	Extensive analysis of the dataset(s) containing clear visuals, observations, and explanations.
	4-5	Good analysis of the dataset(s), containing useful visuals and explanations enough to provide confidence on what is expected.
	2-3	Some analysis has been done, including visuals, but additions work was needed to ensure the dataset is fit for purpose.
	0-1	Not much analysis has been done and the visuals are non-existent or wrong. It is not clear if the dataset is fit for purpose.
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)	2	Comprehensive testing and good explanation of the test results with supporting visuals, which indicates if a model/technique is adequate.
	1	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.
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.
Q5	7-8	Well articulated evaluation of the overall attempt, and great explanation on the choice between the most accurate against the most effective solution.
	4-6	Good explanation on how the original problem is solvable or not and how the experiments helped to understand this.
	2-3	Some evaluation has been done, but not clear explanation on how or if the original problem is solved.
	0-1	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 need of the 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 that needed better explanation.
	0-3	The recording was non-existent or was of poor quality that could not be followed.