Coursework Assignment: Text classification

University of London
BSc in Computer Science
Natural Language Processing

Contents

I. Introduction

This coursework requires you to develop a text classifier and apply it to a specific problem or challenge, e.g. fake news detection, sentiment analysis, spam detection, document tagging, etc. You will need to identify a suitable problem area with an associated data set.

1. Domain-specific area

The first step of the coursework is to identify and describe the problem or challenge. This is an area of industry or science where text classification methods can contribute.

2. Objectives

State and justify the objectives of the project. Discuss its impact and contribution to the problem area. State any contribution which the results may make to the challenge addressed.

3. Dataset

The next step is to identify a suitable dataset which is representative of the challenge and will require attention to all the steps outlined in this assignment. Provide a description of the dataset, its size, data types, the way the data were acquired. State clearly the source of the dataset. Large technology companies, such as Microsoft, Google and Amazon, provide wide variety of datasets.

Example: 'Fake and real news' dataset available from the Kaggle official website.

4. Evaluation methodology

It is good practice in scientific research to decide in advance how you will evaluate the outputs of your investigations. Identify the evaluation metrics you will apply and how they will be applied (e.g. precision, recall, accuracy, F-measure, etc.)

II. Implementation

This part of the coursework is the implementation of the project. It includes and preprocessing the data, building and testing your classifier and obtaining results. The project is expected to be developed using the Python language and Jupyter notebook. Provide well-commented Python code accompanied by document describing the following steps:

5. Preprocessing

Convert/store the dataset locally and preprocess the data. Describe the text representation (e.g., bag of words, word embedding, etc.) and any pre-processing steps you have applied and why they were needed (e.g. tokenization, lemmatization). Describe the vocabulary and file type/format, e.g. CSV file.

6. Baseline performance

Describe and justify the baseline against which you are going to compare the performance of your chosen approach. This can be an already published baseline (e.g. cited in the literature) or the results of a basic algorithm that you implement yourself. The baseline should represent a meaningful benchmark for comparison.

7. Classification approach

Identify any features and labels which will be used in your classifier and justify why they were selected. Build a classifier using an appropriate Python library. Describe your chosen approach, e.g. random forest, support vector machine, Naïve Bayes, logistic regression, etc. and the rationale for selecting it. Run and evaluate your text classifier.

8. Coding style

Your code is expected to meet certain standards as described by accepted coding conventions. This includes code indentation, avoiding unnamed numerical constants and undue use of string literals, assigning meaningful names to variables and subroutines, etc. The code is expected to be fully commented, including variables, sub-routines and calls to library methods.

III Conclusions

9. Evaluation

Evaluate your classifier on the data set. Use the metrics you identified above to quantitatively evaluate the performance of your approach.

10. Summary and conclusions

Provide a reflective evaluation of the project in light of your results. Describe its contributions to the problem area, and discuss the extent to which your solution is transferable to other domain-specific areas. Discuss the extent to which your approach can be replicated by others, e.g. using different programming languages, development environments, libraries and algorithms. Review the potential benefits and drawbacks of alternative approaches.

Rubric

Marks are shown in parentheses.

I. Introduction

1. Introduction to the domain-specific area (200-500 words)
[0] Missing or incorrect
[2] Briefly discussed
[3] Adequately discussed
[4] Domain-specific area clearly stated, informative description, fully referenced work
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

2. Description of the selected dataset (200-500 words)
[0] Missing or incorrect
[2] Briefly described
[3] Adequately described
[4] Described in sufficient details, including origin, size, structure, data types
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

3. Objectives of the project (200-500 words)
[0] Missing or incorrect
[2] Briefly described
[3] Adequately described
[4] Objectives clearly stated with sufficient details and potential contributions
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

4. Evaluation methodology (200-500 words)
[0] Missing or incorrect
[2] Briefly described
[3] Adequately described
[4] Methods and metrics clearly described with convincing rationale
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

II. Implementation

5. Pre-processing
[0] Missing or incorrect
[2] Briefly described
[3] Working code fragments with some pre-processing steps
[4] All appropriate pre-processing steps undertaken, clearly described with convincing rationale

[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

6. Baseline performance
[0] Missing or incorrect
[2] Briefly described
[3] Adequately described, some justification provided
[4] Baseline appropriately chosen, clearly described/implemented with convincing rationale
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

7. Classification approach
[0] Missing or incorrect
[2] Briefly described
[3] Working solution with unconfirmed results
[4] Working solution with confirmed results generated and presented appropriately
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

8. Coding style
[0] Non-meaningful names in code, no comments, use of 'magic numbers'
[2] A minimal attempt at readability
[3] The source code is readable with some comments
[4] The source code is of high quality and follows general coding convention
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

III. Conclusions

9. Evaluation (100 to 300 words)
[0] Missing or incorrect
[2] Briefly described
[3] Results discussed but not convincingly evaluated
[4] Results convincingly evaluated with clear quantitative improvement over suitable baseline
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.

10. Evaluation of the project and its results (200 to 400 words)
[0] Missing or incorrect
[2] Briefly described
[3] Some conclusions without fully evaluating the classifier and the results
[4] Detailed project evaluation (classifier, pre-processing, results, reproducibility)
[5] Exceptional work which includes the above but goes beyond what would be expected from a student at this level.