School of Computing and Information Systems
The University of Melbourne
COMP90049 Knowledge Technologies, Semester 2 2019

Project 2: Geolocation of Tweets with Machine Learning

Released: Friday 20 September 2019

Due: **Part I:** Monday 14 October 2019 - 6AM
Part II: Friday 18 October 2019 - 5PM

Marks: The Project will contribute 20% of your overall mark for the subject.

Overview

For this project, we will be working with tweets that have been annotated by the location of the author. The objective of Part 1 is to build a **geolocation classifier for Tweets**. That is, given a tweet, your system will produce a prediction of where the tweeter is located. For this project, we will limit the set of relevant locations (target classes) to the following three states that are well–represented in the dataset: New York, California, and Georgia.

The goal is to assess the effectiveness of various Machine Learning classification algorithms on the problem of determining a tweeter's location, and to express the knowledge that you have gained in a technical report. This aims to reinforce concepts in data mining and evaluation, and to strengthen your skills in data analysis and problem solving.

The technical side of this project will involve applying appropriate machine learning algorithms to the data to solve the task. You should explore the impact of different features on the performance of the task. The focus of the project will be the report, where you will demonstrate the knowledge that you have gained, in a manner that is accessible to a reasonably informed reader.

Data sets

You will have access to two sets of files:

1. **The RAW Tweets datasets (as a txt file) with the format:**

   ```
   Tweet_id, User_id, tweet text, location (new line)
   ```

2. **The ARFF format datasets:**

   To aid in your initial experiments, we have produced a sample representation of the data in ARFF format (suitable for use with Weka, described below) that you can use. In these files, each instance corresponds to a single tweet, and we have calculated the frequency of some marginally useful terms as attribute values, determined using the following procedure:

   - We pre-processed the training tweets to remove non-alphabetical characters
   - We recorded the term frequency for the top 10, 20, 50, 200 terms according to document frequency
   - We used the method of **Mutual Information** to determine the best 10, 20, 50, and 200 terms

for each of 3 classes (removing the duplicate terms)

The format of the instances in the ARFF file (following the @DATA line in the header) is similar to the familiar **vector space model** in comma-separated value format. For example, the first few attributes are *tweet-id, user-id, atl, atlanta, childplease*. If we observed the following tweet from *Georgia*:

```
161784, USER a_62ef5bb, "Just landed in Atlanta", Georgia
```

The representation in the ARFF file might look like:

```
161784, USER a_62ef5bb,  0,0,0,1,Georgia
```

There is no requirement that you use this data set, but you can use it to start experimenting in Weka straight away.

Tasks

Stage I will comprise the following tasks:

1. **Feature Engineering (optional)**
2. Utilising/Developing **Machine Learning** models to produce a tweet geolocation classifier for the three target cities
3. Writing of a **Report** summarising your findings, analysis, and observations

Stage II will be a peer review process.

Stage I

1. Feature Engineering (Optional)

As discussed in the lectures, the process of engineering features that are useful for discriminating amongst your target class set is inherently poorly-defined. Most machine learning assumes that the attributes are simply given, with no indication from where they came. The question as to which fea- tures are the best ones to use is ultimately an empirical one: just use the set that allows you tocorrectly classify the data.

In practice, the researcher uses their knowledge about the problem to select and construct "good" features. what aspects of a tweet itself might indicate a user's location? You can also find ideas in published papers, e.g., [1].

Attributes typically fall into one of three categories: categorical (*a* or *b* or *c* etc.), ordinal (*cool* < *mild* < *hot*) and numerical (discrete and continuous). All three types can be constructed for the given data. Some machine learning architectures prefer numerical attributes (e.g. *k*-NN); some work better with categorical attributes (e.g. multivariate Naive Bayes) — you will probably observe this through your experiments.

It is **optional** for you to engineer some attributes based on the RAW Tweets dataset (and possibly use them along with the given atributes described below) or you can use the ARFF files generated for you.

2. Machine Learning

Various machine learning techniques have been (or will be) discussed in this subject (Naive Bayes, Decision Trees, 0-R, etc.); many more exist. You are strongly encouraged to make use of machine learning software and/or existing libraries in your attempts at this project.

One convenient framework for this is Weka: http://www.cs.waikato.ac.nz/ml/weka/. Weka is a machine learning package with many classifiers, feature selection methods, evaluation metrics, and other machine learning concepts readily implemented and reasonably accessible. After downloading and unarchiving the packages (and compiling, if necessary), the Graphical User Interface will let you start experimenting immediately.

Weka is dauntingly large: you will probably not understand all of its functionality, options, and output based on the concepts covered in this subject. The good news is that most of it will not be necessary to be successful in this project. A good place to start is the Weka wiki *(http://weka.wikispaces.com/)*, in particular, the primer *(http://weka.wikispaces.com/Primer)* and the Frequently Asked Questions. If you use Weka, please do not bombard the developers or mailing list with questions - the LMS Discussion Forum should be your first port of call.

Some people may not like Weka. Other good packages are available[1][2][3]. One caveat is that you will probably need to transform the data into the correct syntax for your given package to process them correctly. Alternatively, you might try implementing certain components yourself. This will probably be time-consuming, but might give you finer control over certain subtle aspects of the development.

The objective of your learner will be to predict the classes of unseen data. We will use a **holdout** strategy: the data collection has been split into three parts: a **training** set, a **development** set, and a **test** set. This data will be available on the LMS.

1. The **training phase** will involve training your classifier and parameter tuning where required.

2. The **testing phase** is where you observe the performance of the classifier. The development data is labelled: you should run the classifier that you built in the training phase on this data to calculate one or more evaluation metrics to discuss in your report.

3. The **predicting phase**: The test data is unlabeled; you should use your preferred model to produce a prediction for each test instance, and submit your predictions to Kaggle website; we will use this output to confirm the observations of your approach.

To give you the possibility of evaluating your models on the test set, we will be setting up this project on Kaggle InClass competition. You can submit results on the test set there, and get immediate feedback on your system's performance. There is a Leaderboard, that will allow you to see how well you are doing as compared to other classmates participating on-line.

3. Report

You will submit an **anonymised** PDF report, which should describe your approach and observations, both in engineering (optional) features, and the machine learning algorithms you tried.
Your aim is to provide the reader with knowledge about the problem, in particular, critical analysis of the techniques you have attempted (or maybe some that you haven't!). The internal structure of well- known classifiers should only be discussed if it is important for connecting the theory to your practical observations.

Your report should:

- Introduction: Give a short description of the problem and data set
- Literature review: Briefly summarise some relevant literature
- Method: Identify the newly engineered feature(s), and the rationale behind including them (Optional), Explain the methods and evaluation metric(s) you have used (and why you have used them)
- Results: Present the results, in terms of evaluation metric(s) and, ideally, illustrative examples
- Discussion (Critical Analysis): Contextualise the system's behaviour, based on the (admittedly incomplete) understanding from the subject materials
- Conclusion: Clearly demonstrate your identified knowledge about the problem

[1] Orange (http://orange.biolab.si/)
[2] skikit-learn (http://scikit-learn.org/)
[3] PyML (http://pyml.sourceforge.net/)

The critical analysis is key; please think carefully about the following questions:

1. Does your classifier do a good job at addressing the task? Why or why not?

2. Why is the method(s) you explored a reasonable strategy for approaching the task? What advantages does it have over other possible methods?

3. If you engineered new features, why did you use them? What aspect of the data set are they attempting to model?

4. What evaluation strategy did you use? Why? Based on this evaluation, does your model seem to be a good one?

5. Be sure to support your statements and analysis with illustrative examples.

Your report should include a bibliography of relevant or important piece of academic literature. Note that Wikipedia is **not** appropriate as a primary reference (for an overview, see *http://en.wikipedia.org/wiki/Citing_Wikipedia*), although it can occasionally be a good place to start. If you directly use information from a website, e.g., a blog or technical forum, please also be sure to cite those resources.

Note: **the report should be anonymous**, i.e. it should have no mention of your name or student number. *If you have done any implementation and/or feature engineering, you should submit a README that briefly describes how you generate your features (the rationale should be explained in your report), and the purposes of important scripts or external resources, if necessary.*

Stage II (Peer Review)

After the reports have been submitted, there will be a five day period where you will review 2 papers written by your peers in Stage I. The review should be about 200-400 words. In your review, you should aim to have the following structure:

- Briefly summarise what the author has done

- Indicate what you think the author has done well, and why

- Indicate what you think could have been improved, and why

Deliverables

1. An **anonymous** technical report, of 1100-1350 words, submitted to LMS, assessed and formatted as detailed above (by 14 October – 6 AM)

2. The predicted labels of the test tweets submitted to the Kaggle in-class competition described below. (by 14 October – 6 AM)

3. Any code implemented (including scripts and the optional feature engineering) and a README that briefly details your implementation. (optional)

4. Reviews of two research papers written by your peers, each of 200-400 words. (by 18 October - 5PM)

Assessment Criteria

Kaggle performance: (2/20 marks)
The mark for the system ranking will be calculated by first determining the accuracy of each set of predictions for every group. We will then apply equal-frequency binning of the systems in the final system ranking, and assign a score to each group based on the output which occurs in the highest-ranking bin.

Method: (3/20 marks available)
You will identify a knowledge problem, and design experiments using` one or more Machine Learning methods, which could plausibly be used to gain knowledge about that problem. You will describe your method(s) in a manner which would make your work reproducible from your report. You will produce the predicted labels of the test tweets.

Critical Analysis: (8/20 marks available)
You will explain the practical behaviour of your systems, referring to the theoretical behaviour of the Machine Learning methods where appropriate. You will support your observations with evidence, in terms of evaluation metrics, and, ideally, illustrative examples. You will derive some knowledge about the problem of identifying a tweet's location.

Report Quality: (3/20 marks available)
You will produce a formal report, which is commensurate in style and structure with a (short) research paper. You must express your ideas clearly and concisely, and remain within the word limit (1100-1350 words). You will include a short summary of related research.

You can use the marking rubric to indicate what we will be looking for in each of these categories when marking.

Reviews: (4/20 marks available)
You will write a review for each of two reports written by other students; you will follow the guidelines stated above.

Using Kaggle

The Kaggle in-class competition URL will be announced on LMS shortly.

To participate do the following:

- Each student should create a Kaggle account (unless they have one already) using your StudentID
- You may make up to 8 submissions per day. A submission is a comma-separated value (CSV) file with header role "tweet-id, class", with first column represent the test tweet-id values and second column populated with your test predictions of author location {NewYork, Georgia, California}. An example submission file can be found on the Kaggle site.
- Submissions will be evaluated by Kaggle for accuracy, against just 30% of the test data, forming the public leaderboard.
- Prior to competition close, you may select a final submission out of the ones submitted previously – by default the submission with highest public leaderboard score is selected by Kaggle.
- After competition close, public 30% test scores will be replaced with the private leaderboard 100% test scores.

Terms of Use

As part of the Terms of Use of Twitter, in using the data, you must agree to the following:

- The Twitter dataset is based on the data set presented in:

1. Eisenstein, Jacob, et al. A latent variable model for geographic lexical variation. Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2010.
2. Rahimi, Afshin, Trevor Cohn, and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. arXiv preprint arXiv:1804.08049 (2018).

You need to cite these papers in your research paper.

- You are strictly forbidden from re-distributing (sharing) the dataset with others, or re-using it for any purpose other than this project.

Please note that the dataset is a sub-sample of actual data posted to Twitter, with almost no filtering whatsoever. As such, the opinions expressed within the documents in no way express the official views of The University of Melbourne or any of its employees, and my using them does not constitute endorse- ment of the views expressed within. We recognize that some of you may find certain of the documents in bad taste and possibly insulting, but please look beyond this to the task at hand. The University of Melbourne accepts no responsibility for offence caused any content contained in the documents.

Changes/Updates to the Project Specifications

If we require any (hopefully small-scale) changes or clarifications to the project specifications, they will be posted on the LMS. Any addendums will supersede information included in this document.

Academic Misconduct

For most people, collaboration will form a natural part of the undertaking of this project. However, it is still an individual task, and so reuse of ideas or excessive influence in algorithm choice and de- velopment will be considered cheating. We will be checking submissions for originality and will invoke the University's Academic Misconduct policy (http://academichonesty.unimelb.edu.au/policy.html) where inappropriate levels of collusion or plagiarism are deemed to have taken place.

Late Submission Policy

You are strongly encouraged to submit by the time and date specified above, however, if circumstances do not permit this, then the marks will be adjusted as follows:
Each business day (or part thereof) that this project is submitted after the due date (and time) specified above, 10% will be deducted from the marks available, up until 3 business days (1 week) has passed, after which regular submissions will no longer be accepted.

References

[1] Cheng Z., Caverlee J., and Lee K. You are where you tweet: A content-based approach to geo- locating twitter users. In *CIKM'10*, Toronto, Ontario, Canada, 2010. ACM.