School of Computing and Information Systems
The University of Melbourne
COMP90049 Knowledge Technologies, Semester 1 2018

Project 1: waht wierd spelings! aer peaple carzy?

Submission Materials Due:	Code & Report: 3pm (15h00 UTC+10), Wed 11 Apr 2018 (14 marks) Reviews: 9am (15h00 UTC+10), Mon 16 Apr 2018 (4 marks) Secret activity: 9am (15h00 UTC+10), Tue 17 Apr 2018 (2 marks)
Marks:	The project will be marked out of 20, and will contribute 20% of your overall mark for the subject.

Overview

> *rediculous*:
> 1. The alarmingly common misspelling of *ridiculous*.
> 2. To *diculous* again. [sic]

In this project, you will be tasked with comparing and analysing the performance of **spelling correction** methods, on a peculiar data set: a number of headwords taken from UrbanDictionary[1] that have been automatically identified as being misspelled (Saphra and Lopez, 2016). In particular, you must:

- Apply two (or more) spelling correction methods
- Evaluate the behaviour of the methods (for example, using Accuracy), and find some particular headwords that clearly demonstrate the behaviour of each method
- Write a technical report detailing your observations

Resources

You will be given the following resources:

- `misspell.txt`: A list of 716 headwords, one per line, that have been automatically identified as misspelled.
- `correct.txt`: The correct spelling for each of the 716 misspelled headwords, one per line.
- `dictionary.txt`: A list of about 400K tokens from the English language, which is compiled from various sources.

You should note the following: (i) not all of the correct entries appear in the dictionary; (ii) some of the misspelled entries *do* appear in the dictionary; (iii) some of the correct spellings are the same as the "misspelling" (so the word is not misspelled at all!). This is **real-world data**, and it is never guaranteed to be clean. You should attempt to evaluate your approximate matching methods on the data as it is, but you might like to discuss the deficiencies of the datasets in your report.

[1] `http://urbandictionary.com`

Terms of Use

By using this data, you are becoming part of the research community — consequently, as part of your commitment to Academic Honesty, you **must** cite the curators of the dataset in your report, as the following publication:

> Naomi Saphra and Adam Lopez (2016) Evaluating Informal-Domain Word Representations with UrbanDictionary. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, Berlin, Germany. pp. 94–98.

We will flatly **refuse to mark submissions** that plagiarise these authors.

Please note that the dataset is a sub-sample of actual data posted to UrbanDictionary, with almost no filtering whatsoever. Some of the material posted to UrbanDictionary is undoubtedly in poor taste — and though depriving the terms of their context reduces its scope, some of the terms themselves can be construed as offensive. We would ask you to please look beyond this to the task at hand, as much as possible. (For example, it is generally not necessary to actually read the UrbanDictionary postings.)

The opinions expressed within UrbanDictionary in no way express the official views of the University of Melbourne or any of its employees; using the data in a teaching capacity does not constitute endorsement of the views expressed within. The University accepts no responsibility for offence caused by any content contained within this data.

If you object to these Terms, please contact us (`nj@unimelb.edu.au`) as soon as possible.

Report

You will write an **anonymous** report (i.e. no name or student ID, in the header, text or file name) in **PDF format**. This report will detail your analysis, and it should focus mostly on the application of approximate matching methodologies to this problem. This should be a structured technical report, roughly in the style of the sample papers; a sample structure might be as follows:

1. A short description of the problem and data set;
2. An overview of your approximate matching method(s). You can assume that the reader is familiar with the methods discussed in this subject, and instead focus on how they are applied to this task, including some indication of how you determined the parameters (where necessary);
3. A discussion of the effectiveness of the approximate matching method(s) you chose, including a formal evaluation, and some examples to illustrate where the methods were effective/ineffective;
4. Some conclusions about the problem of using approximate matching methods to correct the spelling of UrbanDictionary headwords.

You should include a bibliography and citations to relevant research papers. A good place to begin is probably with the paper cited in the Terms of Use. You might also consider the bibliography from the Approximate Matching lecture slides, in particular:

> Zobel, Justin and Philip Dart. (1996). Phonetic String Matching: Lessons from Information Retrieval. In *Proceedings of the Eighteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*. Zürich, Switzerland. pp. 166–173.

Note that, in general, **you should not cite Wikipedia**. For more information, see the Caution in `http://en.wikipedia.org/wiki/Wikipedia:Citing_Wikipedia` and more generally `http://en.wikipedia.org/wiki/Wikipedia:Academic_use`.

The report should consist of about 1000–1500 words (not including tables/figures and bibliography). This is quite short: you will need to be concise, and you should use tables or graphs to present the data more compactly, where appropriate. You should not discuss the technical details of your implementation, unless they are novel or crucial to your analysis of the methods; you can assume that the reader is familiar with the methods we have discussed in this subject. **Overly long reports will be penalised.**

Note that we will be looking for evidence that you have thought about the task and: have determined reasons for the performance of the methods involved; or can sensibly critique the problem framework. Namely, that you have acquired some **knowledge** that you can supply to the reader. A report that simply records data without corresponding analysis will not receive a strong mark.

We will make report templates (in Rich-Text Format (for MS Word or similar) and LaTeX) available; it would be preferred for you to use these templates when writing your report. Please do not include a title page, abstract, or table-of-contents. We will also post one or more anonymised sample reports, kindly donated by students who have taken this subject in previous years — they are not perfect reports, but received high marks, and might help you to gauge your expectations about what is feasible.

Submission

Submission will entail four parts:

- The code for your approximate matching system(s) and a `README` file which briefly explains how to compile and run your submission and the location and format of the output. Your software can be implemented in any programming language or combination of programming languages.
  You should submit an archive (e.g. ZIP or TGZ) to the "Project 1 Code" link.
- A written report of 1000–1500 words, as a single file in Portable Document Format (PDF).
  This will be submitted via Turnitin, on the LMS to the "Project 1 Report" link.
- Reviews of three papers, which you can access via the "Project 1 Report Reviews" link. For each paper you review, you will have to respond to three "questions":
  - 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

  Each review should be 200–400 words in total.
- The secret task, which you can access via the ~~it's a secret!~~ link.

Marking

The **Report** will be marked according to a marking rubric, for how well it meets three conceptual categories: **Method** (20%), **Analysis** (50%), and **Report Quality** (30%). Each of these will be assessed with a raw mark out of 10 — a mark out of 14 will be obtained by multiplying the raw mark by the given percentage, summing the three categories, and re-scaling to 14 (multiplying by 1.4).

The **Code** will not be directly assessed. Together with the report, this is expected to take 20–25 hours to complete.

For the **Reviews**, 1 mark will be assigned to each completed review, and 1 mark will be assigned for overall effort. Completing the reviews is expected to take about 3–4 hours in total.

You will be assigned 2 marks for completing the **Secret Task**, whcih is expected to take about 1 hour.

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 code or excessive influence in algorithm choice and development will be considered cheating.

In particular, we will be closely comparing the report with submissions made by other students, and with the sample reports. In line with the University's Academic Integrity policies, the report must be wholly written by you, in your own words.

We will invoke the University's Academic Integrity 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 the report is submitted after the due date (and time) specified above, 10% will be deducted from the marks available, up until 5 business days (1 week) has passed, after which regular submissions will no longer be accepted. A late report submission will mean that your report might not participate in the reviewing process, and so you will probably receive less feedback.

- There is no mechanism by which the reviews may be uploaded to the system after the deadline, consequently, it is a major hassle to accept late submissions. Any late submission of the reviews will incur a 50% penalty (i.e. 2 of the 4 marks), and will not be accepted more than a week after the reviewing deadline.

- The secret task will largely be non-sensical to attempt after the deadline. We will reluctantly accept late submissions at a 50% penalty (1 of the 2 marks) up until a week after the task deadline.