School of Computing and Information Systems
The University of Melbourne
COMP90042
WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2017)

Workshop exercises: Week 2

Discussion

1. Give some examples of text processing applications that you use on a daily basis.
2. What is **tokenisation** and why is it important?
   (a) What are **stemming** and **lemmatisation**, and how are they different? Give examples from the `WSTA_N1B_preprocessing` iPython notebook.
3. What is **text classification**? Give some examples.
   (a) Why is text classification generally a difficult problem? What are some hurdles that need to be overcome?
   (b) Consider some (supervised) text classification problem, and discuss whether the following (supervised) machine learning models would be suitable:
      i. k-Nearest Neighbour using Euclidean distance
      ii. k-Nearest Neighbour using Cosine similarity
      iii. Decision Trees using Information Gain
      iv. Naive Bayes
      v. Logistic Regression
      vi. Support Vector Machines
   (c) In the `WSTA_N2_text_classification` iPython notebook, which machine learning model(s) works best on the given classification problem based on the Reuters corpus? Why do you think that is?

Programming

1. Make sure that you have a Python environment where you can run the given iPython notebooks. In particular, ensure that the `numpy`, `sklearn` and `nltk` packages are installed (i.e. you can `import` them).
2. Adapt the `WSTA_N1B_preprocessing` iPython notebook into a program which tokenises a input file based on the five–step model given in the lectures.
3. In the `WSTA_N2_text_classification` notebook, observe how different tokenisation regimes alter the text classification performance of the various classifiers on the given Reuters dataset problem.
   (a) Alter the tokenisation strategy so that it incorporates other stages, for example, punctuation, or stemming/lemmatisation.
   (b) Does performance increase or decrease? Are some classifiers affected more than others? Why do you think that is?

Catch-up

- Revise the following terms, as they are used in a text processing context: "corpus"; "document"; "term"; "token".
- Revise "stop words", and why they are often removed from a text in a text processing/information retrieval context. Use the Web to find a list of stop words for English — are there any words in the list that you might consider not to be a stop word? Are there any words that you consider to be stop words that are missing from the list?
- Recall the most common regular expression operators; practice writing some regular expressions to solve common text processing problems.
- Remind yourself of the difference between the various evaulation metrics discussed in the lectures this week (**accuracy**, **precision**, **recall**, **f-score**). Re-read the (supervised) machine learning pipeline.
- (Re-)familiarise yourself with Python, if you haven't used it recently. In particular, focus on string and array processing, including regular expressions. Also revise functions and mapping mathematical formulae to Python syntax (including the `numpy` package).
- Familiarise yourself with the Natural Language Toolkit (NLTK). You might like to use the e-book `http://nltk.org/book` as a resource; it also covers some of the basics of Python, if you've never used the language before.

Get ahead

- (Extension) Identify some tokenisation issues in a language (other than English) of your choice. How much alteration would need to be made to the tokenisation strategy from the lectures to account for these issues?
- Adjust the `WSTA_N2_text_classification` iPython notebook, so that the supervised machine learning model attempts to solve the **multi-class** problem, rather than the **single-class** problem (for `acq`). Does your assessment of the relative utility of the given classifiers change?