Coursework commentaries 2015–16

CO3354 Introduction to natural language processing

Coursework assignment 1

General remarks

- Throughout this assignment, 'NLTK' refers to the Natural Language Toolkit **version 2**, and 'the NLTK book' refers to *Natural Language Processing with Python* by Steven Bird, Ewan Klein and Edward Loper (2009). Although version 3 of the NLTK is now available, you are recommended to continue using version 2 for compatibility with the course materials. The original version of the textbook is available at www.nltk.org/book_1ed
- You should list all references at the end of your work; these should be properly cited in the text whenever referred to.
- You are required to submit your work as a single PDF file, with an appendix including any Python code and the results of running it.
- Results should be given in the main body of the answer; it is time-consuming to search through an appendix to find answers.
- Some students lost marks by not providing their code in an appendix.

Comments on specific questions

Part 1: Syntax and formal grammars [40 marks]

Question 1

a. Context-free (CF) and regular grammars (RG) in the context of natural language processing.

Answers should include formal differences between the types of rules allowed in regular and CF grammars, ideally with reference to automata, and should note that all regular grammars are CF but not vice versa. Credit may be given for observing that finite state methods are sufficient for many language processing tasks, as detailed in the subject guide. For full marks, answers must reference the 'context of natural language processing' and give examples of natural language constructions which are claimed to require CF-ness. Some students appeared to rely on CS sources which explained the terms in relation to formal or programming languages only. Some said it was 'not possible' to generate, for example, 'if S then S' with RGs; in fact it might be possible to generate some such structures at the cost of generating a lot of unwanted information as well.

There were some good answers but a few students did not attempt this question, and others showed weak understanding. As stated in previous years' reports, this is a topic which many students seem to have trouble with, judging both from coursework answers and performance in the examination. You are advised to revise this topic thoroughly, starting with chapter 2 of the subject guide and references given there.

b. Finite state machines (FSMs).

This question was fairly straightforward and most students did quite well, though some gave away marks by failing to explain their answers. Some answers to (iv) were over-generated. Simply asserting that your grammar is equivalent to the original FSM is not a sufficient answer.

c. These questions involve regular expressions (REs).

Not all students checked correctly for repeating letters or did not use the wordlist as required. Others generated wordlists that clearly did not match the criteria, and showed no awareness of this. Some students referred to code in an appendix, which is fine, but the RE itself should be given as part of the main answer. Most answers were good or very good, but a few showed poor understanding of REs.

Question 2

a. Constructing syntax trees.

This was a straightforward enough question, aimed at giving students some practice at problems they may encounter in the examination. Most achieved full marks.

b. Modifying the grammar to handle various novel constructions, with worked examples.

Answers were of variable quality; some received high marks but others gave incomplete answers, or did not attempt this part. Many students lost marks by not showing worked examples as required. For full marks, the rules should generalise to novel examples and should not over-generate. Some answers were rather ad hoc, tailored to the specific examples in the question, and a few offered rules that did not match their derivations. It can be helpful to look for applications of recursive rules (i.e. rules that include the same nonterminal symbol in the left hand side and the right hand side). Terminal symbols should be single words (e.g. V -> 'will eat' is not acceptable).

Part 2: Corpora and basic text analysis [60 marks]

Question 3

a. Explaining terminology (bookwork).

This was largely a matter of referring to the relevant sections in the subject guide, NLTK book or other sources and paraphrasing their explanations. Most students did fairly well, though some lost marks by failing to give examples as required.

b. Some basic analysis of the texts Dracula and Frankenstein.

The techniques required are presented and explained in the NLTK book. Some students evidently did not read this thoroughly, and made extra work for themselves by manually constructing lists of tokens to be excluded rather than using built-in functionality and the 'stopwords' corpus; others disregarded the requirement to exclude punctuation. Answers (to b. (iv)) tended to be rather brief and impressionistic, lacking detailed reference to the results, though there were some more thoughtful answers. Several students omitted some or all of this subquestion, and/or lost marks by failing to show any code.

Question 4

a. Bookwork question about stemming, requiring examples.

Again the quality of work was very variable: some answers were clear, thorough and well-referenced but many students either omitted this subquestion or gave terse, uninformative or poorly referenced answers. Marks were awarded for technical content, proper referencing, clarity and argumentation – for full marks, each point in the question must be properly addressed. Good answers explained how stemming can improve the performance of text-processing applications, and made a distinction between algorithmic and dictionary-based stemming. Stemmers also differ in how 'aggressive' they are (i.e. how much of the ending of the word they tend to strip off). The NLTK book describes some key differences between the different approaches, and gives examples of when you might choose to use a Porter stemmer or a lemmatiser.

b. A practical problem comparing and discussing the results of different stemmers and a lemmatiser using the same text.

To obtain these results, students needed to follow procedures from the NLTK book, section 3.6, plus a little independent reading for usage of the Snowball stemmer. For full marks, students needed to show evidence of running the code with their results, plausible sets of rules followed by each application, and well-supported answers to question (ii). A few students gave excellent answers, gaining full or close to full marks. Others lost marks by skipping this subquestion, failing to show their code and/or results, or inadequate discussion/motivation of the results.