Coursework commentaries 2016–17

CO3354 Introduction to natural language processing – Coursework assignments 1

General remarks

As in previous years, some students lost marks by ignoring advice given at the start of the assignment.

You should provide citations within the body of your text, and detailed references at the end of your work. Stackoverflow, Wikipedia etc. may give you some ideas, but are not suitable as references. Any answer consisting largely of quotes will not receive many marks. You should **explain your answers in your own words, and show working (where applicable)** for full marks. Each year, students lose marks by omitting explanation or failing to show their working.

You should submit your work as a PDF file, with an appendix including any Python code you have written and the results of running your code. If you have used Jupyter (recommended) you can download your notebook in .ipynb format and submit it as a separate file. Please make sure your code is adequately commented – this can be done using the Markdown option in Jupyter. Some students lost marks by submitting files in other formats.

Comments on specific questions

Question 1: Syntax and formal grammars

a. Students were asked to provide a brief explanation what is meant by **context-free** and **regular** grammars in natural language processing, and the important differences between them. This question, or something very similar, appears regularly in assignments or exams, but many students struggle to give satisfactory answers. The question can be tackled in various ways:

   - **Formally:** both consist of sets of production rules with a single symbol on the left-hand side (LHS). The differences between CFGs and RGs essentially depend on the permitted combinations of terminal and non-terminal symbols on the RHS, and whether they support recursive rules.
   - **Computational properties:** a key criterion is whether the set of strings licensed by a grammar can be processed by a finite-state machine, which can be represented as, for example, a table or a graph, or whether they require a memory or stack such that the system can be in an indeterminate number of states.
   - **Linguistic implications:** certain constructions can only be processed by CFGs, which are arguably also more suited for compositional semantic interpretation.

Whichever approach was chosen, answers should explain relevance to NLP. RGs and CFGs also have applications in general computer science; as in previous years, answers often showed imperfect understanding and/or consisted of partly digested CS-oriented accounts. All of the above points should be supported with appropriate examples.

Students were also asked to modify a sample grammar, and to explain whether context-free or regular grammar rules were more appropriate.

Several different solutions are possible. For full marks, rules should be maximally general and concise, rather than ad hoc and tailored to particular examples; worked examples should also be presented. Some students claimed their grammars were 'regular' although they included CF rules, suggesting their answer to the first part of the question was based on imperfect understanding. Others did not address this part of the question.

a. The following part of the question included a chart which represented a non-deterministic finite state machine (FSM). Most answers showed good understanding and were well-presented. Some answers missed the point that all sentences given in answer to part (iii) must be grammatical.

   Some answers to (iv) were not equivalent to a FSM, again showing that the concept was not fully understood. Some students seemed to miss the requirement for 'equivalent coverage'.

b. The following sub-questions involved regular expressions, and, as in past years, most students were generally confident. Answers to (iv) that matched the required strings but were more verbose than necessary got less than full marks. Some students gave away marks by not providing a list; others coded an RE that over-generated (*e.g*. matched more than one occurrence of particular vowels) but seemed not to notice that their results failed to match the specification. A common error was to omit the ^ at the start and $ at the end.

Question 2: Corpora and basic text analysis

a. This was about word tokenisation, stemming, lemmatisation and sentence segmentation.

   Text normalization is converting text to a standard form or "the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens". See http://nlp.stanford.edu/IR-book/html/htmledition/normalization-equivalence-classing-of-terms-1.html. The required matches may be to search terms, the lexicon used by a parser, 'concepts', base forms of words etc. For example, searches for 'cowboy' may miss tokens of 'cow-boy' or 'Cowboy', while we may want searches for 'terrorism' to also match 'terrorist', 'terrorise', 'terrorize', etc.

   A minimal answer gaining a pass mark would address the following points, with more detailed answers gaining high marks:

   - **Word tokenisation** is identifying substrings in a text that correspond to words: this can involve decisions like, are 'New York', 'isn't', etc. to be treated as one word or two? NLTK's "word tokenizer" confusingly returns punctuation symbols as well as words.
   - **Stemming** is identifying the base form of a word, which may or may not involve simply removing the ending, (e.g. the stem of 'flies' may be returned as 'fly'). Answers should compare different approaches to stemming as exemplified by the Porter and Lancaster stemmers.
   - **Lemmatisation** treats a particular dictionary word as the **lemma** or "base form" of a class of words.
   - **Sentence segmentation** is concerned with finding the sentence breaks in continuous text. In English, the main issue is determining whether or not '.', '?', '!,' etc. actually mark the end of a sentence.

b. required analysis of extracts from *The Virginian* and *The Hound of the Baskervilles*.

   Parts i–iii involve techniques that are described in the NLTK book and/or the subject guide. Some students made it hard for themselves by coding over-elaborate solutions and obtained questionable results. Students who had worked through the course materials should have been well prepared to tackle this question. Some answers to (ii) failed to exclude stopwords and/or punctuation as required, and it should have been an easy matter to spot that something was not working as expected.

   Credit was given for sensible, well-supported answers to (iv). NB the result for 'cowboy' might be 0 unless students normalized 'cow-boy' in the text. Surprisingly few considered this, though several expressed surprise that 'cowboy' was not found. Some changed the query rather than the text, which showed initiative but was not what was asked for.

c. was on the Porter and Lancaster stemmers and the WordNet Lemmatizer. For full marks, students need to show:

   - evidence of running the code together with their results: see Appendix for expected outputs.
   - plausible sets of rules followed by each application – not just lists of words with and without endings.
   - well-supported answers to (ii).

   For (i), students were expected to reconstruct rules on the basis of the results from running the stemmers, not to look them up in other sources or inspect the source code.

   For (ii), students were expected to say why they thought particular results were 'wrong'.

Coursework commentaries 2016–17

CO3354 Introduction to natural language processing – Coursework assignments 2

General remarks

Throughout the second assignment, 'NLTK' refers to the Natural Language Toolkit **version 3**, and 'the NLTK book' refers to *Natural language processing with Python* by Steven Bird, Ewan Klein and Edward Loper, available online at www.nltk.org/book

Students were expected to use NLTK 3 and Python 3. Answers that used NLTK 2 may have had marks deducted.

Question 1: Classification

a. **Bayes' Formula**

Bayes' Formula (or Rule) is fundamental to modern AI and NLP, and regularly comes up in coursework and exam questions. Most students showed confident understanding of this topic and gave good, well presented answers. For full marks, students should explain the answer and show working as stated in the question. Answers lacking explanation or working will be capped; credit will be given for answers that show basic understanding even if the calculations have gone astray.

b. **Movie review classifier**

Students were expected to follow the procedure described in the Supplement to Section 5.4 of the subject guide in the VLE. The key point is that nltk.FreqDist() does not return results in frequency order in NLTK 3 as it did in version 2, but we need to use the most_common() function to find the top 2000 features. Some students overlooked this and so were effectively selecting an arbitrary 2000 features, resulting in poor accuracy. This is why students were advised to read the Supplements first: this point is spelled out in detail in the subject guide notes for Chapter 5. Results may vary on different runs as the corpus is randomly shuffled.

Many of the features students should have found are unsurprising: outstanding, fantastic, memorable, superb have high positive scores while high negatives include poorly, lame, awful, stupid. The presence of certain named actors or subjects seems sufficient to obtain a high score in either direction. A few apparently neutral words are also associated with significant scores.

c. **Test the classifier reviews of the film "La La Land" (2016)**

On examiner runs, both (i) and (ii) were classed as 'pos'. This may indicate the known limitations of a 'bag of words' approach: review (i) contains several terms which appear positive out of context, (e.g. glamour, joy, dream, fun, praised). Credit would be given for discussing how to handle cases like these. Another factor is that the training set includes names of actors and characters who may not appear in these reviews (and vice versa), indicating that a sentiment tool for movies needs to be kept up to date. Finally, the training reviews may have been written in a different, perhaps more informal style than professional reviews. Answers without discussion would achieve limited marks.

NB: the point of this sub-question was to test the classifier that was trained in part (b), not to train a new classifier each time. Since the training procedure involves a random shuffle of the corpus, results will not be directly comparable.

Question 2: Information Extraction

Students were required to extend the chunk grammar from Example 2.3 in Chapter 7 of the NLTK book to handle complex NPs.

NB the question specifies complex NPs; there is no need to define clauses or VPs except as potential constituents of NPs. Explanations should be given in the body of the answer not in the code appendix. The quality of answers varied noticeably: some students gave excellent, carefully thought-out answers, while others simply did not attempt the later parts of the question.

One approach is to use a **cascaded chunker**: the output of one pass will give chunks labelled NP, PP which feed into subsequent passes. This is not the same as a syntactic parser, as a chunker only operates to a finite depth whereas syntactic parsers are (in principle) unbounded. Note that there is not much point in defining single-element chunks as some students did.

Next, students were asked to test their chunker on sentences 200–220 of the tagged WSJ corpus.

Chunking with RE parsers is a **greedy** process which may fail to attach right-frontier PPs or conjuncts at the appropriate level of nesting. Few students noted this: answers should show understanding of this behaviour rather than simply blaming the NLTK for being 'sloppy' as some did. Students were expected to use the pre-tagged and parsed versions of the corpus for testing and comparison, not to tag and (syntactically) parse the original texts themselves.

NB 'Compare' means to describe the results of the chunker, how they differ from the parsed corpus, and try to explain any differences, not just to present a series of examples as if they are self-explanatory.

Question 3: Evaluating Chatbots

In this question, students were asked to compare the performance of the four Loebner prize finalists, addressing issues such as:

- **What can we learn from these examples about the challenges of simulating human interaction?** Challenges include: accessing real-world knowledge, answering questions depending on context, referring back to what has already been said (maintaining a discourse model), reasoning, idiomatic phrases, etc.
- **What particular problems did the higher-scoring systems appear to have solved more effectively?** For example, Mitsuku showed reasonable language understanding, general knowledge (Brexit, colour of bread) and recall of earlier conversation.
- **Why did even the highest scorers still fail to convince the judges they were human?** Mitsuku produced mostly appropriate answers but a couple were just bizarre, enough to give the game away. It stumbled on the reasoning required to resolve reference resolution: Dave, cat questions. None of the chatbots understood 'e4 e5' which is a chess opening (unless 'take on board' is a subtle pun?).

  Credit was given for sensible, well-motivated and answers making the above or similar points, and for correct citation, clarity, structure and argumentation. High-scoring answers will back up their arguments with reference to specific, appropriate examples from the dialogues, rather than generalities. In particular they should discuss questions which involve some kind of reasoning or reference resolution to arrive at an appropriate answer. Many didn't quite get this right: the ability of computer systems to do simple arithmetic or logical inference can hardly be in doubt, the problem is for systems to understand that this is what is needed and how it should be applied.

Several students gave excellent, thoughtful and well-presented answers, though others lost marks by wandering away from the point of the question. Students were specifically asked to compare the performance of the **four finalists**, and were expected to support their answers with reference to specific examples from the dialogues. They were not asked to discuss lower-ranked competitors such as Johnny or Masha, or for general musings on the nature of intelligence and interaction. Given the rather low word limit (though some students failed to reach even the lower bound) there was nothing to spare on irrelevancies. It is not enough to say that a contestant 'made errors' without explaining what these were.