University of London
BSc Computing and Information Systems/Creative Computing
CO3354 Introduction to natural language processing
Coursework assignment 1 2018 – 19

Introduction

- Throughout this coursework 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. This version of the NLTK book is updated for Python 3 and NLTK 3. The first edition of the book, published by O'Reilly, is available at: http://nltk.org/book_1ed/. (There are currently no plans for a second print edition of the book.)
- 'SLP' refers to *Speech and Language Processing* by Daniel Jurafsky & James H. Martin: selected chapters from the draft 3rd edition are downloadable at: https://web.stanford.edu/~jurafsky/slp3/.
- You should list all references at the end of your work, and they should be properly cited in-text whenever referred to. Answers that consist largely of quoted material are unlikely to get high marks, even if properly referenced.
- You should **explain** your answers and show working (where applicable) for full marks. Your main results should be given in the body of your answer rather than relegated to appendices or additional files.
- Please submit your work as **a single PDF file**; this should include Appendices with any Python code you have written and the results of running your code. If you have used Jupyter (recommended) you can additionally download your notebook in .ipynb format and submit it as a separate file (optional).
- Make sure your code is adequately commented – this can be done using the Markdown option in Jupyter. Comments should be grammatical, concise and avoid stating the obvious.
- **Do not upload zip files.**
- Marks may be deducted if you do not submit your work in the required format. There are 100 marks available for this coursework assignment.

Your coursework assignment should be submitted as a single PDF file, using the following file-naming conventions:

YourName_SRN_COxxxxcw#.pdf (e.g. MarkZuckerberg_920000000_CO3354cw1.pdf)

- ***YourName** is your full name as it appears on your student record (check your student portal);*
- ***SRN** is your Student Reference Number, for example 920000000;*
- ***COXXXX** is the course number, for example CO3354; and*
- ***cw#** is either cw1 (coursework 1) or cw2 (coursework 2).*

If you submit an additional Jupyter Notebook file, you should name it following the same convention, with .ipynb in place of .pdf.

REMINDER: It is important that your submitted coursework assignment is your own individual work, and, for the most part, written in your own words. You must provide appropriate in-text citation for both paraphrase and quotation, with a detailed reference section at the end of your assignment (this should not be included in any word count). Copying, plagiarism and unaccredited and wholesale reproduction of material from books or from any online source is unacceptable, and will be penalised (see our guide on how to avoid plagiarism on the VLE).

Preparatory work

Before tackling Question 1, it will be useful to work through the tutorial on English grammar at: www.ucl.ac.uk/internet-grammar/ (IGE). You may notice that this grammar uses some different types of word-classes and grammar rules from the subject guide and the NLTK book: this is a salutary reminder that grammar rules and categories are not fixed and absolute, but are devised by scholars and analysts for specific purposes, on the basis of a limited fragment of the language concerned, and within a particular theoretical framework. Further reading (more advanced): Chapter 10, Formal Grammars of English, from the draft 3rd edition of SLP at: https://web.stanford.edu/~jurafsky/slp3/10.pdf.

Question 2 involves "web scraping", which means extracting the content you are interested in from a webpage and discarding material you have no particular use for. It is recommended that you use BeautifulSoup, a popular Python library for parsing web documents and extracting different kinds of content. We will be primarily interested in extracting text. Section 3.1 of the NLTK book, "Accessing Text from the Web and from Disk", shows how to do this using the get_text() method. However, this can be unsatisfactory if applied to a full document, as it may give you a lot of unwanted material such as markup, scripts and so on. Fortunately, BeautifulSoup enables you to specify which elements you want to extract text from, *e.g.* by limiting the selection to content between paragraph tags. The following snippets show different ways of scraping a page on the Guardian website:

```
from urllib import request
from bs4 import BeautifulSoup
url = "https://www.theguardian.com/politics/2018/sep/20/the-death-of-consensus-how-conflict-came-back-to-politics"
html = request.urlopen(url).read().decode('utf8')
BeautifulSoup(html).get_text()
```

This will give you various kinds of clutter such as control characters, image links etc. Another way is to use the parser, so you only pull out text from selected elements – here we choose title and paragraph:

```
soup = BeautifulSoup(html,'lxml')
for element in soup.find_all(['title','p']):
    print(element.text)
```

You should find that this displays the content of the article more or less as a reader will see it on the screen.

Before tackling Question 2 you are advised to experiment with different ways of scraping text from pages and processing the results. You will need to install as a minimum BeautifulSoup 4 and the 'lxml' parser. There are different ways to do this depending on your platform. Some instructions will be posted to the course area on the VLE. Alongside HTML, PDF is a popular format for distributing documents on the Web, and Question 2 will also require you to process a PDF document. There are various utilities for doing this, but the simplest method is simply to open the file in a standard utility and save it as a text file (removing any editorial matter you don't need). It is recommended that you specify UTF-8 coding both when you save the file and when you open it in Python, otherwise NLTK may crash owing to unrecognised characters.

Question 1

a. The IGE describes three criteria for determining what class a word belongs to:
   - Meaning
   - Shape or form of the word
   - Position/environment in a sentence.

   The genre of 'nonsense verse' pioneered by Edward Lear and Lewis Carroll features made-up words which do not have any recognised meaning, though we can usually say which word class they would belong to (some of Carroll's coinages have in fact entered the language). Read through the verse from Carroll's Jabberwocky below and explain how you would decide which classes to assign to *brillig, slithy, tove, gyre, gimble, wabe, mimsy, mome, raths, outgrabe*.

   Twas brillig, and the slithy toves
   Did gyre and gimble in the wabe:
   All mimsy were the borogoves,
   And the mome raths outgrabe

   [10 marks]

b. Run the whole text of Jabberwocky through the NLTK POS tagger (nltk.pos_tag()), normalised to lower case and using the default Penn Treebank tagset. Are there any cases where you think the tagger has made an incorrect choice? If so, list the first 10 of these from the beginning of the poem, and discuss likely reasons for the tagger's decisions.

   The Penn Treebank tagset is documented at: www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html and you can find the text of Jabberwocky at various web locations including the Project Gutenberg text of Carroll's Through the Looking Glass: www.gutenberg.org/files/12/12-h/12-h.htm.

   [15 marks]

c. Read through the sections of the IGE on Word Classes, Introducing Phrases, and Clauses and Sentences, and the relevant sections of the subject guide and the NLTK book before answering the following questions:
   i. Following the IGE definitions of word classes and phrase types, construct a set of formal grammar rules that will generate the example phrases listed below, using a similar format to the grammar on p.22 of the subject guide. Draw syntax trees for examples (A) and (G). You may ignore punctuation and capitalisation.

      (A) The children in class 5 play football in the park across the road
      (B) The waitress gave me the wrong dessert
      (C) Two of my guests have arrived in a taxi
      (D) Louis deliberately broke the window
      (E) My sister is fond of animals
      (F) Fatima graduated very recently

(G) If you give your details to our secretary, we will contact you when we have a vacancy

ii. Explain whether your rules make up a **context-free** or **regular** grammar.

[25 marks]

Question 2

This question involves textual comparisons of two articles of a similar length, which are both concerned with populism in contemporary politics:

- "The death of consensus: how conflict came back to politics" by Andy Beckett, Guardian 20 September 2018: www.theguardian.com/politics/2018/sep/20/the-death-of-consensus-how-conflict-came-back-to-politics
- "The rise of populism and the crisis of globalisation: Brexit, Trump and beyond" by Michael Cox, Irish Studies in International Affairs, 2017. http://eprints.lse.ac.uk/86880/7/Cox_Rise%20of%20populism%20published_2018.pdf

This question follows on from Chapter 1, Section 3 of the NLTK book, which involves using simple statistical analysis of the words in a text to get an idea of its topic and genre. The point of this exercise is to investigate whether the inclusion of semantically related terms, as identified by WordNet, can assist in this enterprise.

a. Explain the following terms in your own words, and using your own examples:
   i. Synonym
   ii. Antonym
   iii. Hypernym
   iv. Hyponym
   v. Meronym
   vi. Holonym

[9 marks]

b. List the following for both documents:
   i. Collocations.
   ii. 24 most common words, normalised to lower case and excluding punctuation, stopwords and words of less than 5 characters.

[8 marks]

c. Tabulate a conditional frequency distribution for occurrences of the words *Brexit, anger, parties, people, political, power, populist, Trump* in the two documents.

[8 marks]

d. Read through at least Section 5 of Chapter 2 in the NLTK book and the HOWTO page at: www.nltk.org/howto/wordnet.html.

Using the results from part b(ii) above:

i. List all words in the lists which occur in both documents;
ii. Using Wordnet, generate the synsets for all words in both lists, and give all the lemmas which occur in the results for both documents. How does this compare with your results for (i.) above?
iii. Generate the hypernyms for all words in both lists, and give those that occur in both sets of results. How much does this list on its own tell you about the content and subject matter of the original articles?
iv. Repeat (i-iii) using the 50 most common words from each document, with the same exclusions. Do you find the results to be more, or less informative?

[25 marks]

[Total: 100 marks]

[END OF COURSEWORK ASSIGNMENT 1]