# COMP34412

Coursework 2

# Task 1a – NER

a) NLTK provides a classifier that has been trained to recognise several types of named entities (see Section 5 at http://www.nltk.org/book/ch07.html). Use the function nltk.ne_chunk( ) to process corpus 1 (see Data below).

(1a) Corpus 1 processed with NLTK:

**0 / 0.25 / 0.5 points**

# Task 1b – NER

b) Use the Stanford named-entity recogniser ( https://nlp.stanford.edu/software/CRF-NER.shtml) to process the same corpus.

(1b) Corpus 1 processed with Stanford NER:
**0 / 0.25 / 0.5 points**

# Task 1c – NER

*c) Compare the outputs of the two NER methods for the ORGANIZATION class. Explore the differences between the two approaches – which of the tools seems better in getting the bounders of named entities right? Provide some examples. In how many cases the tools fully agree between themselves on the mentions of named entities (exact match), and in how many cases they have a partial overlap. Your report should be around 1 page long.*

(1c) Boundary detection discussed with examples

**0 / 0.5 / 1 point**

# Task 1c – POS tagging

*c) Compare the outputs of the two NER methods for the ORGANIZATION class. Explore the differences between the two approaches – which of the tools seems better in getting the bounders of named entities right? Provide some examples. In how many cases the tools fully agree between themselves on the mentions of named entities (exact match), and in how many cases they have a partial overlap. Your report should be around 1 page long.*

(1c) Agreement between tools discussed

**0 / 0.5 / 1 point**

# Task 2a – sentiment analysis

a) *With the popularity of social media, building and maintaining a sentiment/polarity lexicon is a huge challenge. In the first part of this task, you will build a sentiment lexicon using a semi-supervised approach by bootstrapping the process, starting from a small lexicon of adjectives (see at the end of the document) and corpus 2 (see Data below). Write a program that will collect more adjectives to populate the lexicon and assign them with the likely polarity. For example, adjectives conjoined by "and" are likely to be of the same polarity (e.g. corrupt and brutal), while adjectives conjoined by "but" are not (fair but brutal).*

*continues*

# Task 2a – sentiment analysis

a) *From previous page... Consider (and implement) other possible patterns; consider how you could assign a polarity if an adjective appears several times in the corpus (and, for example, you have conflicting polarity signals). Evaluate the outcome – how many of the proposed adjectives have been properly classified (according to your judgment of their typical polarity)? Provide explanations for any errors in the report (~1 page).*

# Task 2a – sentiment analysis

(2a) Patterns designed for finding new ADJ (e.g. relying on POS, taking into account polarity, frequency, etc.):

**0 / 1 / 1.5 points**

(2a) Additional patterns implemented, conflicts resolved

**0 / 0.5 / 1 point**

(2a) Evaluation of the produced dictionary

**0 / 0.25 / 0.5 points**

# Task 2b – sentiment analysis

*b) The two files in corpus 2 represent positive and negative examples of movie reviews. As a baseline, implement a classifier that simply counts whether there are more positive or negative words in a review (use the MPQA lexicon Subjectivity Cues Lexicon, see below).*

(2b) Baseline implemented: **0 / 0.5 / 1 point**

# Task 2b – sentiment analysis

*b) ⋯ Then build a machine-learning sentiment classifier using a simple bag-of-words approach. Expand the feature set to include other possible features: e.g. whether any of the words in the review come from a polarity lexicon (e.g. MPQA), whether they are negated, etc.*

(2b) Features: simple BoW: **0 / 0.25 / 0.5 point**

(2b) Extra features: lexicon, negation, etc.:
**0 / 0.5 / 1 point**

# Task 2b – sentiment analysis

*b) For training your classifier, use any available machine-learning framework (e.g. scikit-learn or Weka). Evaluate the method using k-fold cross validation and compare to the baseline. Explain briefly what you have done and discuss the results in the report (~1 page).*

(2b) Classifier + k-fold cross-validation:

**0 / 0.5 / 1 point**

(2b) Comparison/discussion:

**0 / 0.25 / 0.5 points**

# General comments

- Don’t forget to submit code!
- Please add your name to the report and make it look professional