0.1 Exercise 1

This is the answer to question one on the tutorial sheet. The raw text I wrote to challenge the tokeniser was:

Sentences like ‘Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo‘ which demonstrate lexical ambiguity are ones I’d 100% expect text parsers to have difficulty with. Phrases involving colloquialisms from the R.O.I’d also be tough to intepret like ‘what’s the craic’, ‘I’m absolutely stuffed’, or ‘thats gas’; all are Irish expressions.

I used this text first because the sentence ’Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo’ is syntactically ambiguous yet grammatically correct and the high word-repetition involving different meanings of the word buffalo (the animal, location,verb) would prove challenging for text parsers as it requires the grammatical structure to parse correctly. In addition, the text contains mispellings, contractions, a variety of punctuation, and Irish colloquial phrases which might prove difficult to text analytics. The use of 40% was used as the usage of the percent symbol alters what ’40’ refers to which I would expect a text parser to have trouble interpretting correctly.

1. Load the file in and use nltk.word_tokenizer() on it. Report the list of tokens that are produced from it and note any oddities that arise. Comment on these oddities and how they might be handled.

   - tokens = [’Sentences’, ’like’, ’‘’, ’Buffalo’, ’buffalo’, ’Buffalo’, ’buffalo’, ’buffalo’, ’buffalo’, ’Buffalo’, ’buffalo’, ’‘’, ’which’, ’demonstrate’, ’lexical’, ’ambiguity’, ’are’, ’ones’, ’I’, ’’’, ’d’, ’100’, ’%’, ’text’, ’parsers’, ’to’, ’have’, ’difficulty’, ’with’, ’.’, ’Phrases’, ’involving’, ’colloquialisms’, ’from’, ’the’, ’ROI’, ’’’, ’d’, ’also’, ’be’, ’tough’, ’to’, ’intepret’, ’like’, ’‘’, ’what’, ’’’, ’s’, ’the’, ’craic’, ’’’, ’,’, ’‘’, ’I’, ’’’, ’m’, ’absolutely’, ’stuffed’, ’’’, ’,’, ’or’, ’‘’, ’thats’, ’gas’, ’’’, ’;’, ’all’, ’are’, ’Irish’, ’expressions’, ’.’]

     The key oddities which are present in the tokens are:

   - Punctuation - The tokens includes punctuation such as full stops, quotation marks, semi-colons, apostrophes, etc. To resolve this, the data should be pre-processed to strip common punctuation marks.
   - Contractions - Contractions in the text such as "ROI’d" (an informal "Republic of Ireland would") are split by word tokenise. While this is the correct behaviour of Word Tokeniser (per here) this may not always be desired and might be difficult to interpret. As the number of contractions is typically small, expanding this out into the full term as a pre-processing step would be feasible. Alternatively, one could use an alternative word tokeniser which does not split contractions such as TweetTokenizer.
   - Digits and Percentages - As punctuation marks are split 40% is split into ’40’ and ’%’. This might be problematic as percentage might be described

using many different words (the symbol, the word, etc.) and digits might be described in words and digits. Also, 40% might be misintereprетted as the integer 40 instead of 40 out of 100.

2. Now, take this output from the tokenizer and do the normalization step.
   - My normalisation consisted of
     a) lower casing the text
     b) replacing a number of contractions
     c) removing punctuation
     d) swapping digits with the text using num2words
     e) unifying the word percent, percentages, %.
     f) Removing stopwords.
   - tokens = ['sentences', 'like', 'buffalo', 'buffalo', 'buffalo', 'buffalo', 'buffalo', 'buffalo', 'buffalo', 'buffalo', 'demonstrate', 'lexical', 'ambiguity', 'ones', 'would', 'one', 'hundred', 'percent', 'expect', 'text', 'parsers', 'difficulty', 'phrases', 'involving', 'colloquialisms', 'roi', 'would', 'also', 'tough', 'intepret', 'like', 'craic', 'absolutely', 'stuffed', 'thats', 'gas', 'irish', 'expressions']

3. Now, take the output from normalization step and run it through a pos-tagger. Report this output as your answer and highlight any inaccuracies that occur at this stage.
   - Unsurprisingly the key block of innaccuracies occur in the lexically ambiguous sentence as it fails to capture the true sequence of 'PN N ON N V V PN N', while other failures include incorrectly categorising nouns as verbs, some remaining contractions as verbs, and proper noun abbreviations (ROI) as a single noun
   - pos_tagged=[('sentences', 'Noun, plural'), ('like', 'Preposition or subordinating conjunction'), ('buffalo', 'Noun, singular or mass') , ('buffalo', 'Noun, singular or mass'), ('buffalo', 'Noun, singular or mass') , ('buffalo', 'Noun, singular or mass'), ('buffalo', 'Noun, singular or mass') , ('buffalo', 'Noun, singular or mass') , ('buffalo', 'Noun, singular or mass') , ('buffalo', 'Noun, singular or mass'), ('demonstrate', 'Verb, base form'), ('lexical', 'Adjective'), ('ambiguity', 'Noun, singular or mass'), ('ones', 'Noun, plural'), ('would', 'Modal'), ('one', 'Cardinal number'), ('hundred', 'Cardinal number'), ('percent', 'Noun, singular or mass'), ('expect', 'Verb, non-3rd person singular present'), ('text', 'Noun, singular or mass'), ('parsers', 'Noun, plural'), ('difficulty', 'Verb, non-3rd person singular present') , ('phrases', 'Noun, plural'), ('involving', 'Verb, gerund or present participle'), ('colloquialisms', 'Noun, plural'), ('roi', 'Noun, singular or mass') , ('would', 'Modal'), ('also', 'Adverb'), ('tough', 'Adjective'), ('intepret', 'Noun, singular or mass'), ('like', 'Preposition or subordinating conjunction'), ('whats', 'Noun, plural') , ('craic', 'Verb, non-3rd person singular present') , ('absolutely', 'Adverb'), ('stuffed', 'Adjective'), ('thats', 'Noun,

plural') , ('gas', 'Noun, singular or mass'), ('irish', 'Adjective'), ('expressions', 'Noun, plural')]

0.2 Exercise 2

This is the answer to question two on the tutorial sheet. The raw text I wrote to challenge the tokeniser was:

"Brown Thomas and Arnotts (BTA) are too Irish companies, operating 7 physical shops, which were purchased by the Selfridges Group in 2014. BT&A sell quality brands and host class events which're some craic for shopaholics, but their stores are a teensy bit extravagent (at least as my friend Thomas says)."

I used this text because it contains a number of business names in a variety of formats (Brown Thomas and Arnotts, BTA, Selfridges Group, BT&A), a name which can easily be confused with the business, years in digit format, numbers in digit format, mispellings, and non-standard dictionary terms due to its highly informal phrasing. These aspects I believe would be challenging for text analytics.

1. Tokenize the new-text-file (50words) and the stem it using Porter Stemming. Report your outputs and some of weird things that Porter Stemming does.
   - stemmed_tokens=['brown', 'thoma', 'and', 'arnott', '(', 'bta', ')', 'are', 'too', 'irish', 'compani', ',', 'oper', '7', 'physic', 'shop', ',', 'which', 'were', 'purchas', 'by', 'the', 'selfridg', 'group', 'in', '2014', '.', 'bt', '&', 'a', 'sell', 'qualiti', 'brand', 'and', 'host', 'class', 'event', 'which', "'re", 'some', 'craic', 'for', 'shopahol', ',', 'but', 'their', 'store', 'are', 'a', 'teensi', 'bit', 'extravag', '(', 'at', 'least', 'as', 'my', 'friend', 'thoma', 'say', ')', '.']
   - We see it removes the s from a variety of non-plural nouns ('Thomas'), removes the 'y' from quality and replaces it with an 'i' and liekwise in teensy, changes extavagent to extravag, operating to opera, and shopaholic to shopahol. In general for the above text, it removes es and s from the ends of words, removes ant from words, turns y at the end of words into i, and removes 'al' from the ends of words. It removes the Proper Noun component of a number of words (e.g. 'Brown' in 'Brown Thomas')

2. Tokenize this, the new-text-file, and then lemmatize it using WordNet Lemmatizer; note you may have to pos-tag the sentences first and then convert the tags to make this work. Report the result of these steps and point out some of the things that look wrong
   - stems=['Brown', 'Thomas', 'and', 'Arnotts', '(', 'BTA', ')', 'are', 'too', 'Irish', 'company', ',', 'operating', '7', 'physical', 'shop', ',', 'which', 'were', 'purchased', 'by', 'the', 'Selfridges', 'Group', 'in', '2014', '.', 'BT', '&', 'A', 'sell', 'quality', 'brand', 'and', 'host', 'class', 'event', 'which', "'re", 'some', 'craic',

'for', 'shopaholic', ',', 'but', 'their', 'store', 'are', 'a', 'teensy', 'bit', 'extravagent', '(', 'at', 'least', 'a', 'my', 'friend', 'Thomas', 'say', ')', '.']

- We see it shortens 'as' to 'a' which is incorrect.

3. Compare the outputs from Porter Stemming and the Lemmatisation of the same file. Which do you think is the best to use and why?

   - The key areas where Porter and Wordnet differ from the original words or each other are:

```
+-------------+----------+-------------+
| Original    | Porter   | Wordnet     |
+=============+==========+=============+
| Brown       | brown    | Brown       |
+-------------+----------+-------------+
| Thomas      | thoma    | Thomas      |
+-------------+----------+-------------+
| Arnotts     | arnott   | Arnotts     |
+-------------+----------+-------------+
| BTA         | bta      | BTA         |
+-------------+----------+-------------+
| Irish       | irish    | Irish       |
+-------------+----------+-------------+
| companies   | compani  | company     |
+-------------+----------+-------------+
| operating   | oper     | operating   |
+-------------+----------+-------------+
| physical    | physic   | physical    |
+-------------+----------+-------------+
| shops       | shop     | shop        |
+-------------+----------+-------------+
| purchased   | purchas  | purchased   |
+-------------+----------+-------------+
| Selfridges  | selfridg | Selfridges  |
+-------------+----------+-------------+
| Group       | group    | Group       |
+-------------+----------+-------------+
| BT          | bt       | BT          |
+-------------+----------+-------------+
| A           | a        | A           |
+-------------+----------+-------------+
| quality     | qualiti  | quality     |
+-------------+----------+-------------+
| brands      | brand    | brand       |
+-------------+----------+-------------+
| events      | event    | event       |
+-------------+----------+-------------+
| shopaholics | shopahol | shopaholic  |
+-------------+----------+-------------+
| stores      | store    | store       |
+-------------+----------+-------------+
| teensy      | teensi   | teensy      |
+-------------+----------+-------------+
| extravagent | extravag | extravagent |
+-------------+----------+-------------+
| as          | as       | a           |
+-------------+----------+-------------+
| Thomas      | thoma    | Thomas      |
+-------------+----------+-------------+
| says        | say      | say         |
+-------------+----------+-------------+
```

- I believe the correct process to use is highly dependent on the context. While PorterStemming is fast, we can see it produces a number of key errors in how it stems words by employing highly generalised rules. WordNet appears to more correctly account for context in its grouping, however it is slower as it needs to check the pos tags and the pos_tagging may in some cases be innaccurate as we've seen in 3, however it is likely to be more accurate. While I believe WordNet is the better approach to use in general, there may be scenarios where performance rather than accuracy may be a key criteria (such as when dealing with high-volume streaming data) in which case it may be acceptable/better to use PorterStemming.