**Q1: Try to get used to using then nltk package. Create a text file with,50 words in it (written as a collection of sentences) and make sure it has items that challenge the tokeniser (e.g., I.B.M. and other weird things). But, you need to make these up yourself, so you think about what it is doing.**

**a) 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.**

The result using the word tokenizer on my file is as follows:

['Conversely', **','**, 'it', 'has', 'been', 'argued', 'that', 'these', 'huge', 'leaps', 'in', 'NLP', 'were', 'made', 'as', 'a', 'result', 'of', 'a', 'focus', 'shift', 'from', **'“'**, 'rationalist', **'”'**, 'methods', 'to', **'“'**, 'empirical', **'”'**, 'or', **'“'**, 'corpus-based**', '”'**, 'methods', **'('**, 'Brill', '&', 'Mooney', **',**', '1997', **',**', **'p.', '1', ')', '.'**, 'The', 'former', 'detailing', 'a', 'very', 'hand-coded', 'method', 'which', 'required', 'a', 'lot', 'of', 'self-observation', 'as', 'opposed', 'to', 'the', 'empirical', 'method**', '.'**]

Please see table below reporting on the oddities and how they might be handled.

| Oddities that arise | How oddities might be handled |
|---|---|
| Punctuation marks such as ‘,’, '“', '(', ')', '.' have all been included. | Carrying out the normalisation step to replace punctuation marks. |
| In text citations are treated as separate words. 'p.', '1' | Pre-process the citations to remove the space between characters. |

**b) Now, take this output from the tokenizer and do the normalization step.**

During the normalization step, I attempted to convert a token to its base form. This was done by changing the text to lowercase, replacing punctuation marks and removing stop words. The output after normalisation was carried out can be seen below:

['conversely', 'argued', 'huge', 'leaps', 'nlp', 'made', 'result', 'focus', 'shift', 'rationalist', 'methods', 'empirical', 'corpus-based', 'methods', 'brill', '&', 'mooney', '1997', 'p', '1', 'former', 'detailing', 'hand-coded', 'method', 'required', 'lot', 'self-observation', 'opposed', 'empirical', 'method']

**c) 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.**

After I took the output of from the normalization step and ran it through a pos-tagger, the output can be seen below:

[('conversely', 'RB'), ('argued', 'VBD'), ('huge', 'JJ'), ('leaps', 'NNS'), **('nlp', 'RB')**, ('made', 'VBD'), ('result', 'NN'), ('focus', 'NN'), ('shift', 'NN'), ('rationalist', 'NN'), ('methods', 'NNS'), ('empirical', 'JJ'), ('corpus-based', 'JJ'), ('methods', 'NNS'), ('brill', 'NN'), ('&', 'CC'), ('mooney', 'NN'), ('1997', 'CD'), ('p', 'NN'), ('1', 'CD'), ('former', 'JJ'), ('detailing', 'VBG'), ('hand-coded', 'JJ'), ('method', 'NN'), ('required', 'VBN'), ('lot', 'RB'), ('self-observation', 'NN'), ('opposed', 'JJ'), ('empirical', 'JJ'), ('method', 'NN')]

| **Inaccuracies that Occurred:** |
|---|
| 1. “nlp” is described as an adverb when is should be a noun |

There was only one inaccuracy observed in this piece of text and that comes from an abbreviation. It would be quite difficult for to correctly identify “nlp” as Natural Language Processing and understand that this is a noun.

**Q2:**

**a) Tokenize a new-text-file (50 words) and the stem it using Porter Stemming. Report your answer and some of weird things that Porter Stemming does.**

The new text file has been tokenized and stemmed using Porter Stemming. The result is as follows:

[('The', 'the'), **('initial', 'initi')**, ('phase', 'phase'), (',', ','), ('tokenization', 'token'), **('involves', 'involv'),** ('breaking', 'break'), ('string', 'string'), ('into', 'into'), ('tokens', 'token'), ('that', 'that'), ('can', 'can'), ('be', 'be'), ('used', 'use'), ('.', '.'), ('Formally', 'formal'), (',', ','), ('it', 'it'), ('can', 'can'), ('be', 'be'), **('defined', 'defin')**, ('as', 'as'), ('“', '“'), **('defines', 'defin')**, ('a', 'a'), **('commonly', 'commonli')**, ('accepted', 'accept'), ('standard', 'standard'), ('of', 'of'), ('clitic', 'clitic'), ('tokenization', 'token'), (':', ':'), **('separating', 'separ')**, ('conjunctions', 'conjunct'), (',', ','), **('affixival', 'affixiv')**, ('prepositions', 'preposit'), ('and', 'and'), ('pronouns', 'pronoun'), (',', ','), **('future', 'futur'),** ('marker', 'marker'), ('clitics', 'clitic'), (',', ','), ('and', 'and'), **('definite', 'definit')**, ('articles.', 'articles.'), ('”', '”'), ('(', '('), ('Diab', 'diab'), (',', ','), ('2009', '2009'), (',', ','), ('p.', 'p.'), ('2', '2'), (')', ')'), ('In', 'in'), ('layman', 'layman'), ('’', '’'), ('s', 's'), ('terms', 'term'), (',', ','), ('it', 'it'), ('could', 'could'), ('be', 'be'), ('thought', 'thought'), ('of', 'of'), ('as', 'as'), ('a', 'a'), ('kind', 'kind'), ('of', 'of'), ('delimiter', 'delimit'), ('.', '.')]

Some oddities that can be observed with Porter Stemming are can be seen in the table below.

| Oddities Observed |
|---|
| 1. Converting “initial” to “initi” |
| 2. Converting “involves” to “involv” |
| 3. Converting “defined” to “defin” |
| 4. Converting the last “y” to “i”. Example, converting “commonly” to “commonli” |
| 5. Converting “separating” to “separ” |
| 6. Converting “affixival” to “affixive” |
| 7. Converting “future” to “futur” |
| 8. Converting “definite” to “definit” |

**b) Tokenize the new-text-file and then lemmatize it using the WordNetLemmatizer; 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.**

The new text file has been tokenized and lemmatized using WordNetLemmatizer. Pos tagging was necessary first and then converting the tags using a dictionary. The result can be seen below:

[('The', 'n'), ('initial', 'a'), ('phase', 'n'), (',', 'n'), **('tokenization', 'n'),** ('involve', 'v'), ('break', 'v'), ('string', 'v'), ('into', 'n'), ('token', 'n'), ('that', 'n'), ('can', 'n'), ('be', 'v'), ('use', 'v'), ('.', 'n'), **('Formally', 'r')**, (',', 'n'), ('it', 'n'), ('can', 'n'), ('be', 'v'), ('define', 'v'), **('a', 'n')**, ('“', 'a'), ('define', 'v'), ('a', 'n'), **('commonly', 'r')**, **('accepted', 'a'),** ('standard', 'n'), ('of', 'n'), ('clitic', 'a'), **('tokenization', 'n')**, (':', 'n'), **('separating', 'n')**, ('conjunction', 'n'), (',', 'n'), ('affixival', 'a'), ('preposition', 'n'), ('and', 'n'), ('pronoun', 'n'), (',', 'n'), ('future', 'a'), ('marker', 'n'), ('clitics', 'n'), (',', 'n'), ('and', 'n'), ('definite', 'a'), **('articles.', 'n')**, ('”', 'n'), ('(', 'n'), ('Diab', 'n'), (',', 'n'), ('2009', 'n'), (',', 'n'), ('p.', 'v'), ('2', 'n'), (')', 'n'), ('In', 'n'), ('layman', 'n'), ('’', 'n'), ('s', 'n'), ('term', 'n'), (',', 'n'), ('it', 'n'), ('could', 'n'), ('be', 'v'), ('think', 'v'), ('of', 'n'), **('a', 'n'),** ('a', 'n'), ('kind', 'n'), ('of', 'n'), **('delimiter', 'n')**, ('.', 'n')]

Some things that appear wrong with this method can be seen in the table below.

| Inaccuracies using WordNetLemmatizer |
|---|
| 1. “tokenization” remained as “tokenization” and should have been turned to “token” |
| 2. “formally” should have been turned into “formal” |
| 3. “as” got converted to “a” |
| 4. “commonly” should have been converted to “common” |
| 5. “accepted” should have been converted to “accept” |
| 6. “separating” should have been converted to “separate” |

| |
|---|
| 7. When tokenization does not work effectively, the lemmatizer cannot detect the word properly. An example is “articles.”, which still includes the punctuation mark. The lemmatizer did not work and the word remained “articles.” |
| 8. “delimiter” should have been converted to “delimit” |

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

It appears both Porter Stemming and Lemmatisation have their positives and negatives and coincidentally, I found exactly 8 faults in both methods.
Porter Stemming appears to be extremely fast but inaccurate, it works by cutting off the end or the beginning of the word. This indiscriminate cutting can be successful on some occasions, but not always. For example, please see a couple of examples where oddities were observed using this method below.

| Form | Stem |
|---|---|
| initial | initi |
| involves | involv |
| defined | defin |

Lemmatisation looks beyond word reduction and considers a language’s full vocabulary. This can be seen when the lemma of “was” became “be”. However, this method is much slower but more accurate. As seen from my inaccuracies table, it can still make mistakes such as converted “as” to “a”.

Overall, if I was to suggest which method to use, I would suggest using Lemmatisation. The trade off between speed and accuracy is worth it in my opinion. However, this would rely on the requirement of the individual themselves. If speed is the requirement of the user, Porter Stemming would be the best option for them.