

Tutorial 3 – Tagging, Stemming, and Lemmatisation

Consider the following text:

I am going on mountains. I will climb till I can't no more.

1. Create a new python script
2. Put the text above into a variable
3. Tokenise the text using nltk's word tokeniser
4. Normalise the cases (to lower case)
5. Remove stop words
6. Now it's time to tag each token with its appropriate parts of speech tag. Assuming the tokens are in a list called *tokens*,

```
pos_tags = nltk.pos_tag(tokens)
print(pos_tags)
```

7. Observe the output
8. Can you find 3 words from the text that need to be stemmed/lemmatised? And to what?
9. We will start with stemming. We are going to use the Porter Stemmer which is implemented in nltk.

```
porter = PorterStemmer()
for token in tokens:
    stemmed_token = porter.stem(token)
    print(token + " -> " + stemmed_token)
```

10. In order for lemmatisation to work we will use of a lemmatiser called WordNet Lemmatiser (part of the nltk library). Therefore, we need a mapping function that can map the Parts of Speech tags from the current format to the Wordnet format. We will use the function below, that is very widely used by the NLP community.

Add the following imports to the script.

```
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet
```

And this just in case:

```
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
nltk.download('punkt')
```

And the function:

```
def get_part_of_speech_tags(token):
    #We are focusing on Verbs, Nouns, Adjectives and
Adverbs here.
    tag_dict = {"J": wordnet.ADJ,
                "N": wordnet.NOUN,
                "V": wordnet.VERB,
                "R": wordnet.ADV}

    tag = nltk.pos_tag([token])[0][1][0].upper() #eg. from
[('She', 'PRP')] get P
    return tag_dict.get(tag, wordnet.NOUN)
# if no match, return Noun as the POS
```

At the end of the script:

```
lemmatizer = WordNetLemmatizer()

for token in tokens:
    lemmatisedtoken = lemmatizer.lemmatize(token,
    get_part_of_speech_tags(token))
    print(token, lemmatisedtoken)
```

11. Observe the output. Which words were stemmed/lemmatised?