**COMP6713 - Natural Language Processing – 25 T1**

Tutorial – Week 2

**Installing NLTK:** <https://www.nltk.org/install.html>

**Installing SpaCy:** <https://spacy.io/usage>

**Installing Transformers:** <https://huggingface.co/docs/transformers/en/installation>

We strongly recommend that you use this tutorial to install jupyter notebooks, to write the code for the below. We will be providing jupyter notebooks for the future tutorials.

**Q. 1.** Compare and analyse the output of NLTK and spaCy POS taggers for the following sentences:

1. The cat is on the mat

2. "Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo" is a grammatically correct English sentence. It is often used as a stress test for POS taggers. [[1](https://en.wikipedia.org/wiki/Buffalo_buffalo_Buffalo_buffalo_buffalo_buffalo_Buffalo_buffalo)]

3. “The old man the boat”

Are the POS taggers only looking up words to return POS tags? Or does the output change when you try slight modifications of the above sentences? Try different examples to explore.

What errors do the POS taggers make? Try other examples.

**Q. 2.** Masked language modelling (MLM) is a form of self-supervision. A dataset of unlabeled sentences can be converted to a self-supervised MLM task as follows: Randomly mask a word in a sentence. The input to the language model is the sentence with the word masked, the output is the word at the masked position in the original sentence.

The sentences in the dataset are:

1. You can never be overdressed or overeducated.
2. Always forgive your enemies; nothing annoys them so much.
3. I am not young enough to know everything.
4. Draw the black-box view to create a self-supervised MLM task as described above.
5. Use the HuggingFace pipeline ‘fill-mask’ to obtain the top 5 words for each of the sentences above.
6. Calculate precision@5. Recall from the lecture that precision is the number of correctly retrieved words. “@5” indicates that any of the top 5 words need to be correct.

**Extra:**

**Q. 3.** Use the spaCy matcher code provided in the class and extend it to also return the WordNet lemmas of words that are extracted as entities.