**TEXT PROCESSING (2 points / 20 minutes)**

**PARSING (3 points / 30 minutes)**

**LANGUAGE MODELLING (2 points / 20 minutes)**

1. What is a language model, that is, what is it's purpose in terms of probabilities ? (0.5 points)

2. If is a bigram, which is the meaning of this equation ?

Explain what are , how to compute , and the meaning of probability being computed. (0.5 points)

3. One of the applications of a language model is spell checking. We have seen there are two types of spell checking, named *real word* and *non-word*. What is the difference, which one is harder and why ? (0.5 points)

4. How can a language model help in real word spelling correction ? In order to answer it, first explain the meaning of the problem formulation

In particular, what is and how is it computed ? Which one of the two probabilities is computed by the language model ?

Hints: are sentences and is the typed one, are the prior and likelihood, respectively. (0.5 points)

**Validation questions for the language model exercise**

1. In the first part of the exercise you were asked to try three variants of n-gram language models. Which one was *not* one of them ?

a) MLE

b) EveryGram

c) Laplace

d) StupidBackoff

2. Second part of the exercise was about training a small feedforward neural network to learn a language model. Once we had trained the network we could generate sentences word by word, by feeding the network with the last few generated words. The network output was then, for each word in the vocabulary, the probability of being the next word, and we had to choose one. Two strategies were possible, to be implemented with numpy’s argmax and torch.multinomial. Brielfy explain the idea of these two strategies, in one or two lines each.

3. Last part of the exercise was about real word spell correction with a language model. Was this an n-gram model like in part 1 or a model learned by the neural network of part 2 ?

**SEQUENCE LABELLING (3 points / 30 minutes)**