# NLP course Assignment 3: Grammar Engineering

Daniel Bazar 314708181

Lior Krengel 315850594

## Part 1 – Weights

1. **Why does the program generate so many long sentences? Specifically, what grammar rule is responsible for that and why? What is special about this rule? discuss.**The program generates so many long sentences mainly because of some recursive grammar rules.   
   Specifically, the rule: NP → NP PP is responsible for that phenomena. Since every rule in this part has the same weight, and we start from the rule S → NP VP, then, we have 50% chance of choosing the above rule (because there are two rules for NP). In addition, the VP rule is also derived to a rule that includes NP. Moreover, the nonterminal PP that is derived from the NP rule contains in itself the nonterminal NP which means that NP is called at least twice.
2. **The grammar allows multiple adjectives, as in: “the fine perplexed pickle”. Why do the generated sentences show this so rarely? discuss.**

Generated sentences show multiple adjectives so rarely because in order to get such an event we need to derive the rule Noun → Adj Noun twice. This rule is one of 6 rules with Noun in the LHS, and all rules have the same weight, so the probability for it is just 1/6.

1. **The grammar format allows specifying different weights for different rules. Which numbers must you modify to fix the problems in (1) and (2), making the sentences shorter and the adjectives more frequent? Verify your answer by generating from the grammar. Discuss your solution (which rules did you modify, and why).**

First, to make sentences shorter we need to handle the problem with the NP → NP PP rule we described in (1). The problem was that this rule is chosen too often. so we increased the weight of the second NP rule (NP → Det Noun) to 5 making the relative ratio between the two rules in favor of the “non-recursive” one.

Second, to make multiple adjectives more frequent, we took the same approach, and we increased the weight of the Noun → Adj Noun rule to 5 to make it more frequent.

1. **What other numeric adjustments can you make to the grammar in order to favor a set of more natural sentences? Experiment and discuss.**

In order to make sentences more natural, we have adjusted the weights of the rules. This adjustment reflects the likelihood of word distribution. To estimate this, we have adjusted the weights of verbs and adjectives based on the nouns they are commonly paired with. For each verb or adjective, we have assigned a weight based on how many nouns can appear with it in the same context. For instance, for the verb 'ate', we assigned the weight 2 because it can connect with two nouns out of five (one can 'ate' a sandwich and a pickle, but not a president, a chief of staff, or a floor). Similarly, we determined the weights of the nouns based on the verbs and adjectives they are commonly paired with. We didn't change the weights of the determiners and prepositions because they are likely uniformly distributed.   
We have also adjusted the weights of the rules used to create full sentences. We increased the weight of a sentence ending with a period (".") because it is more likely than sentences ending with an exclamation mark ("!") or a question mark ("?").

## Part 2 | Extending the Grammar

We made modifications to the grammar in order to generate the types of phenomena illustrated in the given sentences. All explanations can also be seen in the “grammar2” file.

For each sentence, we provide the corresponding modifications:

* 1. NP → Nnp – to support proper nouns (people, locations, organization, etc.)
  2. Nnp → Sally – adding “Sally” to vocabulary as Nnp
  3. NP → NP Cc NP – to support coordinating conjunction
  4. VP → Verb Cc VP – to support conjunction between verbs
  5. Cc → and – adding “and” to vocabulary as Cc

To respect the *subcategorization frame of verbs* we reclassified the ‘Verb’ preterminal as transitive verbs (V1) and we added another preterminal called ‘V0’ as intransitive verbs.

* 1. VP → V0 – to support intransitive verb phrase
  2. V0 → sighed – adding “sighed” to vocabulary as V0