# COMP9318

# Project1 Bonus

**Group Name:** Twosome
**Group Member:** GUANQUN ZHOU, z5174741 KAIWEN LUO, z5100899

### Significantly outperforms our implementation.(Bonus)

- In Q3, we found that Kneser-Ney smoothing is probably the appropriate method to solve the problem. So we decided to optimize our code and algorithm.
- The formula of Kneser-Ney smoothing is like this:

$$P_{KN}(w_i|w_{i-1}) = \frac{\max(C(w_{i-1}w_i) - d, 0)}{C(w_{i-1})} + \lambda(w_{i-1})P_{continuation}(w_i)$$

where

$$\lambda(w_{i-1}) = \frac{d}{\mathrm{C}(w_{i-1})} \cdot |\{w: C(w_{i-1}, w) > 0\}|$$

$$P_{continuation} = \frac{N(.w_i)}{N(..)}$$

and

$$N(.w_i) = |\{w_{i-1}|c(w_{i-1}, w_i) > 0\}|$$
$$N(..) = |\{(w_{i-1}, w_i)|c(w_{i-1}, w_i) > 0\}|$$

- The difference is in Q3, the value of $d$ (discounting ) we choose is 0.75 which is the widely recognized empiric value.

  In bonus part, after optimizing the algorithm, the expression of $d$ is $d = \frac{n_1}{n_1+2n_2}$.

  * $n_i$ represents the number of n-gram that occur $i$ times

  Similarly, for UNK, we set $N(.w_i) = 1$, which is the best guess.
- In this way, we can outperform our implementation and the result is 113. The margin is raised to 21, which is larger than 17.

### The instruction of how to execute our code.

- All the code should be implemented in python3.
- We already combined all the functions we need together, so our code can be executed as the way to execute the implement for Q1,Q2 and Q3
  Like this:

```
import submission as submission
bonus_result = submission.Bonus(State_File, Symbol_File, Query_File)
```