Universität des Saarlandes
Prof. Dr. Dietrich Klakow
Lehrstuhl für Signalverarbeitung
SNLP Summer Term 2025

Exercise Sheet 6

Backing-off Language Models

Deadline: 28.05.2025 23:59

Guidelines: You are expected to work in a group of 2-3 students. While submitting the assignments, please make sure to include the following information for all our teammates in each of your PDF files/python scripts:

Name:

Student ID (matriculation number):

Email:

Your submissions should be zipped as **Name1_id1_Name2_id2_Name3_id3.zip** when you have multiple files. For assignments where you are submitting a single file, use the **same naming convention** without creating a zip. For any clarification, please reach out to us on the **CMS Forum**. These instructions are mandatory. If you are not following them, tutors can decide not to correct your exercise.
Please note:

- Ex 6.1 and 6.2 are written assignments, please submit a pdf (written using Latex) with the **names, matriculation IDs and emails** of all team members for this part. In case you are not familiar with Latex, clearly written handwritten submissions are also accepted, but we strongly encourage pdfs written using Latex.
- Ex 6.3 and 6.4 are programming assignments, you can write your code in the supplied notebooks and submit them. Don't forget to put in your **names, matriculation IDs and emails** in the given sections.
- Submit the pdfs and notebooks together in a zip file in CMS. No need to resubmit any datasets or pycache.

Exercise 6.1 - Smoothing and Interpolation (0.5+0.5=1 points)

We talked about discounting algorithms in the previous assignment. Here, we will compare them with smoothing algorithms.

a) Additive (Laplace) smoothing is a classic technique for ensuring that every possible event (such as a word in a language model) receives some nonzero probability, even if it was never observed in the training data. For unigrams, the add-α smoothed probability for word w_i is:

$$P_{add-\alpha}(w_i) = \frac{C(w_i) + \alpha}{N + \alpha|V|}$$

, where $C(w_i)$ is the count of w_i in the corpus, N is the total number of tokens, $|V|$ is the vocabulary size, and $\alpha > 0$ is the smoothing parameter.

Now, what distribution would you get if you applied add-α smoothing infinitely? e.g. if F_{smooth} is a function that smooths a language model using add-α smoothing and $\text{lm}^{(n+1)} = F_{\text{smooth}}(\text{lm}^{(n)})$. What will the language model $\lim_{n\to\infty} \text{lm}^{(n)}$ look similar to? Explain your reasoning.

b) Another commonly used smoothing method is linear interpolation, explain how it works in 3-4 sentences, and also show the formula for this.

Exercise 6.2 - Kneser-Ney smoothing (1+0.5+0.5=2 points)

One of the more popular methods for smoothing language models is Kneser-Ney smoothing. It makes use of *continuation counts* of words for lower order n-grams, given as

$$C_{KN} = \begin{cases} \text{count}(\bullet) & \text{for highest order} \\ \text{continuationcount}(\bullet) & \text{for lower orders} \end{cases} \tag{1}$$

For a trigram distribution, Kneser-Ney Smoothing is implemented using the following equations:

$$P_{KN}(w_3|w_1, w_2) = \frac{\max\{N(w_1 w_2 w_3) - d, 0\}}{N(w_1 w_2)} + \lambda(w_1, w_2) P_{KN}(w_3|w_2)$$

$$P_{KN}(w_3|w_2) = \frac{\max\{N_+(\bullet w_2 w_3) - d, 0\}}{N_+(\bullet w_2 \bullet)} + \lambda(w_2) P_{KN}(w_3)$$

$$P_{KN}(w_3) = \begin{cases} \frac{N_+(\bullet w_3)}{N_+(\bullet\bullet)} & \text{if } w_3 \in \text{V} \\ \frac{1}{V} & \text{otherwise} \end{cases} \tag{2}$$

λ is used to normalize the discounted probability mass and is given by

$$\lambda(w_1, w_2) = \frac{d}{N(w_1 w_2)} \cdot N_+(w_1 w_2 \bullet)$$

$$\lambda(w_2) = \frac{d}{N(w_2)} \cdot N_+(w_2 \bullet)$$

Now, based on the given information, answer the following questions:

a) Understand what these terms represent and fill it in the table given here (4-5 words each).

Kneser-Ney term	Description
$\text{N}(\text{w}_1 w_2 w_3)$	
$\text{N}(\text{w}_1 w_2)$	
$\text{N}_+(\bullet w_2 w_3)$	
$\text{N}_+(\bullet w_2 \bullet)$	
$\text{N}_+(\bullet w_3)$	
$\text{N}_+(\bullet\bullet)$	
$\text{N}_+(w_1 w_2 \bullet)$	
$\text{N}_+(w_2 \bullet)$	
$\lambda(w_1 w_2)$	
$\lambda(w_2)$	

b) Assuming a general corpus which has an equivalent distribution as the English language, how will Kneser-Ney Smoothing handle these bigrams: "Abu Dhabi", "Game Over"? (answer in 3-4 sentences)?

c) What are the advantages of Kneser-Ney over Good-Turing smoothing? Answer in 2-3 sentences.

Exercise 6.3 - Linear Interpolation and cross-validation (1.5 + 1= 2.5 points)

See attached notebook

Exercise 6.4 - Kneser-Ney vs the World (0.5 + 0.5 + 1 + 0.5 + 0.5 + 1 + 0.5 = 4.5 points)

See attached notebook