UNIVERSITÄT
DES
SAARLANDES

Statistical Natural Language Processing

Summer Semester 2019
Prof. Dr. Dietrich Klakow

Exercise Sheet VI

Submission Deadline: June 7th, 23:59

1 Advanced Language Modelling

Exercise 1.1: Kneser-Ney Language Models (10 points)

In this exercise you will derive the backing-off factors for a trigram Kneser-Ney LM. You are not required to implement Kneser-Ney smoothing but to explore and understand how it differs from previously introduced smoothing techniques.

Consider the following notation

- $\mathcal{V}$ - LM vocabulary
- $N(x)$ - count of the n-gram x in the training corpus
- $N_{1+}(\bullet\, w) \triangleq |\{u : N(u, w) > 0\}|$ - number of bigram types ending in w
- $N_{1+}(w\, \bullet) \triangleq |\{u : N(w, u) > 0\}|$ - number of bigram types starting with w
- $N_{1+}(\bullet\, w\, \bullet) \triangleq |\{(u, v) : N(u, w, v) > 0\}|$ - number of trigram types with w in the middle

For a trigram Kneser-Ney LM, the probability of a word w_3 is estimated as

$$\begin{aligned}
\mathbf{P}_{KN}(w_3|w_1w_2) &= \frac{\max\{N(w_1w_2w_3) - d, 0\}}{\sum_{v\in\mathcal{V}} N(w_1w_2v)} + \lambda(w_1w_2)\mathbf{P}_{KN}(w_3|w_2) \qquad (1)\\
&= \frac{\max\{N(w_1w_2w_3) - d, 0\}}{N(w_1w_2)} + \lambda(w_1w_2)\mathbf{P}_{KN}(w_3|w_2) \qquad (2)
\end{aligned}$$

As shown above, the definition of the highest order probability in Kneser-Ney LM is identical to that in absolute discounting. However, the main difference is in estimating the lower-order back-off probabilities

$$\begin{aligned}
\mathbf{P}_{KN}(w_3|w_2) &= \frac{\max\{N_{1+}(\bullet\, w_2w_3) - d, 0\}}{\sum_{v\in\mathcal{V}} N_{1+}(\bullet\, w_2v)} + \lambda(w_2)\mathbf{P}_{KN}(w_3) \qquad (3)\\
&= \frac{\max\{N_{1+}(\bullet\, w_2w_3) - d, 0\}}{N_{1+}(\bullet\, w_2\, \bullet)} + \lambda(w_2)\mathbf{P}_{KN}(w_3) \qquad (4)
\end{aligned}$$

The base case of this recursive definition is the unigram probability which is estimated as

$$\mathbf{P}_{KN}(w_3) = \frac{N_{1+}(\bullet\, w_3)}{\sum_{v\in\mathcal{V}} N_{1+}(\bullet\, v)} = \frac{N_{1+}(\bullet\, w_3)}{N_{1+}(\bullet\, \bullet)} \qquad (5)$$

where $N_{1+}(\bullet\, \bullet)$ is the number of bigram types. Contrary to absolute discounting, a Kneser-Ney LM does not interpolate with the zerogram probability. In case w_3 was not in the vocabulary of the LM (i.e., OOV), then $\mathbf{P}_{KN}(w_3|w_1w_2) = \mathbf{P}_{KN}(w_3) = \frac{1}{|\mathcal{V}|}$.

(a) Assume that $N(w_1w_2) > 0$, and use the facts $\sum_{v\in\mathcal{V}} \mathbf{P}_{KN}(w_3 = v|w_1w_2) = 1$ in addition to $\sum_{v\in\mathcal{V}} \mathbf{P}_{KN}(w_3 = v|w_2) = 1$, to derive the back-off factors of a trigram Kneser-Ney LM; $\lambda(w_1w_2)$ and $\lambda(w_2)$.
Hint: start by taking the sum over the vocabulary for both sides of Equation (2) when deriving $\lambda(w_1w_2)$ and Equation (4) when deriving $\lambda(w_2)$.

(b) From the `corpus.sent.en.train` corpus, collect the necessary statistics to/and fill the table below

	w = '`york`'	w = '`matter`'
$N(w)$		
$N_{1+}(\bullet\, w)$		
$-\log_2 \mathbf{P}_{ML}(w)$		
$-\log_2 \mathbf{P}_{Lids}(w)$		
$-\log_2 \mathbf{P}_{KN}(w)$		

where $\mathbf{P}_{ML}(w)$ is maximum likelihood estimation of the unigram probability (no smoothing), $\mathbf{P}_{Lids}(w)$ is the Lidstone-smoothed unigram probability with $\alpha = 1$, and $\mathbf{P}_{KN}(w)$ is the Kneser-Ney unigram probability defined in Equation (5). Given the information in the (filled) table, discuss your observations.

(c) Explain the main intuition behind Kneser-Ney language models and discuss how they are different than other back-off language models (that is, absolute discounting LMs).

Your solution should include the intermediate steps for the derivation in part (a), the table and the source code of part (b), and a sufficient explanation of part (c).

2 Text Classification

Exercise 2.1: Author Identification using LMs (10 points)

In this exercise, you will investigate whether language models can be used to identify the author of a text fragment. The text corpora in the compressed file `exercise6_corpora` consist of literary works of three different authors; Charles Dickens, Arthur Doyle, and Mark Twain. Your task is to train a LM for each author and attempt to identify the author of a (held out) text fragment where the identity of the author is unknown.

(a) For each corpus in the `train` folder, preprocess the text and extract the sentences using NLTK sentence tokenizer.

(b) From each preprocessed corpus, extract unigrams and bigrams (using the `word_ngrams` function and the provided regex-based tokenizer to tokenize each corpus). Collect unigram and bigram counts and store them in a suitable data structure.

(c) Generate a table of the top 15 frequent unigrams and bigrams for each author with their relative frequencies. Do you observe any noticeable differences between authors?

(d) For each author, build a unigram LM and a bigram LM. You could either use Lidstone smoothing or absolute discounting smoothing as long as your implementation of the model is correct and has been tested in previous assignments.

(e) For each text fragment in the `test` folder, apply the same prepossessing steps that were applied to the training corpora.

(f) Compute the perplexity of each test fragment using each unigram LM in part (c). Generate a 3×3 table to show the results. Use Google search engine to get the actual author of each test fragment and discuss your observations.

(g) Repeat part (f) but this time use the bigram LMs. Write your observations.

(h) Discuss why the LMs succeed/fail in this task.

Your submission should include the source code and the tables of parts (c), (f), and (g).

Exercise 2.2: Feature Selection (Extra credits)

The goal of this exercise is to use the point-wise mutual information (PMI) measure to explore the most discriminating n-grams for each author. The PMI measure has not been introduced in the lecture yet but the concept is based on the fundamentals of information theory. PMI is defined as

$$\mathrm{PMI}(f, c) = \log_2 \left[\frac{\mathbf{P}(f, c)}{\mathbf{P}(f)\mathbf{P}(c)} \right]$$

where f is the feature (in this exercise, the word n-gram) and c is the category (in this exercise, the author), $\mathbf{P}(f, c)$ is the probability of observing feature f within category c, and $\mathbf{P}(f)$ is the probability of observing feature f across all categories. Both $\mathbf{P}(f, c)$ and $\mathbf{P}(f)$ can be estimated using maximum likelihood from the training data. Using the unigram and bigram counts collected from previous exercise, proceed in this exercise as follows

(a) Write a function that computes and returns the PMI value of an n-gram feature f and a category c.

(b) For the unigram features that have occurred more than 15 times across all training corpora in exercise 2.1, compute PMI between these features and each author. (Assume that $\mathbf{P}(c)$ is uniformly distributed across authors).

(c) Generate a table of the top 10 unigram features for each author (highest positive PMI value). Briefly discuss your observations.

(d) Repeat parts (b) and (c) for the bigram features. Briefly discuss your observations.

Your solution should include the tables of part (c) and (d), as well as the source code to reproduce the results. Depending on the clarity of your presentation and the novelty of your solution, you may earn up to 5 points for this exercise.

Submission Instructions

The following instructions are mandatory. Please read them carefully. If you do not follow these instructions, the tutors can decide not to correct your exercise solutions.

- You have to submit the solutions of this exercise sheet as a team of 2-3 students.
- Besides NLTK sentence tokenizer, NLTK modules are not allowed, and not necessary, for this assignment.
- You do not need to include the distributed corpora within your submission.
- Make a single `ZIP` archive file of your solution with the following structure

  - A `source_code` directory that contains your well-documented source code and a `README` file with instructions to run the code and reproduce the results.
  - A `PDF` report with your solutions, figures, and discussions on the questions that you would like to include.
  - A `README` file with group member names, matriculation numbers and emails.

- Rename your `ZIP` submission file in the format

  `exercise06_id#1_id#2_id#3.zip`

  where `id#n` is the matriculation number of every member in the team.

- Your exercise solution must be uploaded by only one of your team members to the course management system (CMS). Once the grading is done, your assignment grade will be distributed to each team member by one of the tutors.

- If you have any problems with the submission, contact `babdullah@lsv.uni-saarland.de` before the deadline.