SNLP 2017
Exercise 5

Submission date: 02.06.2017, 23:59

Smoothing

1) (3 points) In most backing-off models, the smoothing distribution is usually fixed to be a less-specific distribution i.e $\beta(w|\hat{h}) = p(w|\hat{h})$ with $\hat{h}$ being the history h shortened by one word h' i.e $h = \hat{h}h'$. However, in some cases, it is better to use a dedicated backing-off distribution.
Given an absolute discounting model,

$$p(w|h) = \begin{cases} \frac{N(w,h) - d}{N(h)} + \alpha(h)\beta(w|\hat{h}), & \text{if } N(w,h) > 0 \\ \alpha(h)\beta(w|\hat{h}), & \text{otherwise} \end{cases} \tag{1}$$

and back-off weight

$$\alpha(h) = \frac{d}{N(h)} N_+(h) \tag{2}$$

where,

$$N_+(h) = \sum_w N_+(wh) \tag{3}$$

derive an expression for $\beta(w|\hat{h})$ such that the marginal distribution of the resulting joint distribution $p(w, h'|\hat{h})$ is identical to $p(w|\hat{h})$:

$$p(w|\hat{h}) = \sum_{h'} p(w, h'|\hat{h}) = \sum_{h'} p(w|h)p(h'|\hat{h}) \tag{4}$$

Hint: Substitute (1) into (4), and assume maximum likelihood estimation for $p(w|\hat{h})$ and $p(h'|\hat{h})$. The expression you are to derive is in (slide 5, page 50).

2) (2 points) Given the following statistics:

count, $N(w,h)$	count of Counts, $n_{N(w,h)}$
1	5000
2	1600
3	1000
4	600
5	300

Table 1: These are all the bigrams with the history "wine"

count, $N(w,h)$	bigram
1	wine drinker
2	wine lover
3	wine glass

a) What are the discounted counts under GoodTuring discounting for the three given bigrams?

b) The amounts from discounting counts are given to a back-off unigram model. Using such a back-off model, what are the probabilities for the following bigrams?
(i) $p(drinker|wine)$
(ii) $p(glass|wine)$
(iii) $p(mug|wine)$
Note: $p(mug) = 0.015$, $p(drinker) = 0.015$, $p(glass) = 0.01$. State any assumptions that you make.

Cross-Validation

3) (5 points) In this exercise, you will implement absolute discounting model and learn how to tune the discounting parameter d using K-fold cross validation. Use the text file in the materials folder for this task.

a) (2 points) Implement a bigram language model with absolute discounting using discounting parameter d to smooth the bigram and unigram distributions.

$$P(w|h) = \begin{cases} \frac{N(w,h)-d}{N(h)} + \alpha(h)P(w) & \text{if } N(w,h) > 0 \\ \alpha(h)P(w) & \text{else} \end{cases} \tag{5}$$

$$P(w) = \begin{cases} \frac{N(w)-d}{N} + \alpha\frac{1}{V} & \text{if } N(w) > 0 \\ \alpha\frac{1}{V} & \text{else} \end{cases} \tag{6}$$

If the history h of the bigram (w, h) is not found in the training corpus return the estimate of the unigram language model $P(w)$.[1]
A recommended approach of solving this task is:

* (1 point) tokenize the text, and compute the unigram counts, bigram counts and $R(h)$ for all unigrams. (see slide 5, page 21)
* (1 point) create a function that returns $P(w|h)$ when given parameters d, $N(w, h)$, $N(h)$ and $R(h)$.

b) (3 points) In the lecture you learned about leaving-one out cross validation. This sometimes can lead to a soultion in a calculation on paper but is often not suitable for numerical calcluations. Instead of leaving only one word out for cross validation, you will implement a function where you always leave out a different fraction of the data for validation. This method is called K-fold cross validation, where K stands for the number of fractions (or folds as they are called).
Split the data into K parts. Use each of the K parts once for validation and merge the rest for training (i.e estimation of bigram distribution in 3a). Calculate perplexity on the validation set. Take the average of the K perplexity values and return it.

* (0.5 point) create a function that returns perplexity given relative frequencies of the test bigrams and training bigram distribution.
* (1.5 point) implement a 5-fold cross-validation function that takes d as an input. The function should return on output the average of 5 perplexity values.
* (1 point) Plot the cross-validation perplexity for different $d \in [0.1, 0, 2, .., 1.0]$. Don't forget to label the axis. What is the optimal value for d?

1 Submission Instructions: Read carefully

- You can form groups of maximum 2 people.
- Submit only 1 archive file in the ZIP format with name containing the MN of all the team members, e.g.:

Exercise_01_MatriculationNumber1_MatriculationNumber2.zip

- Provide in the archive:
  - your code, accompanied with sufficient comments,
  - a PDF report with answers, solutions, plots and brief instructions on executing your code,
  - a README file with the group member names, matriculation numbers and emails,

[1] A language model is defined for a specific history. In case a history is unseen an arbitrary other language model can be used. This is called a fall back language model.

  - Data necessary to reproduce your results [2]

- The subject of your submission mail must contain the string [SNLP] (including the braces) and explicitly denoting that it is an exercise submission, e.g:

  [SNLP] Exercise Submission 01

- Depending on your tutorial group, send your assignment to the corresponding tutor:

  - David Adelani: *s8daadel@stud.uni-saarland.de*
  - Rajarshi Biswas: *rbisw17@gmail.com*
  - Xiaoyu Shen: *s8xishen@stud.uni-saarland.de*

2 General Information

- In your mails to us regarding the tutorial please add the tag [SNLP] in the subject accompanied by an appropriate subject briefly describing the contents.
- Feel free to use any programming language of your liking. However we strongly advise in favor of Python, due to the abundance of available tools (also note that Python3 comes with an excellent native support of UTF8 strings).
- Avoid using libraries that solve what we ask you to do (unless otherwise noted).
- Avoid building complex systems. The exercises are simple enough.
- Do not include any executable files in your submission, as this may cause the e-mail server to reject it.
- **In case of copying, all the participants (including the original solution) will get 0 points for the whole assignment. Note: it is rather easy to identify a copied solution. Plagiarism is also not tolerated.**
- **Missing the deadline will result in 0 point for the whole assignment. So make sure to submit on time.**
- **Please submit in your solutions necessary to support your claims. Failure to do so, might results in reduction of points in the relevant questions.**
- Each assignment has 10 points and perhaps some bonus points (usually 2 or 3). In order to qualify for the exams, you need to have 2/3 of the total points. For example, in case there are 12 assignments, you need to collect at least 80 out of the 120 points to be eligible for the exams. A person that gets 10 plus 2 bonus points in every exercise, needs to deliver only 7 assignments in order to be eligible for the exams, since 7*12=84.
- Attending the tutorial gives 1 point increase for the corresponding assignment.
- Presenting a solution will give bonus equal to the point of the presented question (Up to 4 poins). The bonus is divided evenly in the group member. Each group can present up to 3 times.
- Exercise points (including any bonuses) guarantee only the admittance to the exam, however have no further effect on the final exam grade.

[2] If you feel that these files are beyond reasonable size for an email submission and also reasonably convenient, please provide a means for us to access them online