MOOC on Approaches to Machine Translation: Rule-based, Statistical and Hybrid

MT-MOOC

http://mooc.upc.edu

Universitat Politècnica de Catalunya

Assignment 2 (Week 3): Finding the intra-coherence of the Gospels

Goal

The main goal of this exercise targets to make you comfortable with the concepts and use of Language Modelling and additionally help you to understand the usefulness of discounting, back-off and interpolation. In order to do that we will play around some famous similar text that are feasible to our needs: The Gospels

Requirements

The exercise is intended to be done in any operating system: Windows / Mac OS X and Linux. You will need SRI Language Modeling tookit in order to complete the assignment:

SRILM (from SRI International): You can download and compile it at `http://www.speech.sri.com/projects/srilm/`.

We provide three different zip packages with different compiled versions of the toolkits. Along the the compiled resources we provide a script (guide.bat in Windows and guide.sh in Linux/Mac OS) that may guide you in how to use external resources for obtaining the necessary data. ***By downloading them you agree that you will restrict their use only for MT-MOOC learning purposes.***

Windows (tested on Windows 8.1) :

http://mt-mooc.upc.edu/lti/includes/problems/win32.zip

Linux (tested on Ubuntu 12.04) :

http://mt-mooc.upc.edu/lti/includes/problems/linux.zip

Mac OS X (tested on Yosemite) :

http://mt-mooc.upc.edu/lti/includes/problems/macosx.zip

For any problem, please post a new discussions at the discussions part of the MOOC. We also recommend the of an advanced text editor for doing your programming such as:

Sublime Text Editor : `http://www.sublimetext.com/`

Notepad++ : `http://notepad-plus-plus.org/`

Estimated time: 2 hours. Maximum 6 hours

Deadline: Nov 25th at 23:59:59 CET

Description

We are going to continue our expertise within the world of Bible studies, in that case we are going to make use of our excellent knowledge about Language Modelling. Among the files contained in the zip file, you have found:

Assignment.pdf : The current assignment description

ngram(.exe) and ngram-count(.exe): Compiled programs from SRILM that you need to use

sed.vbs (Only Windows) : sed program to extract information from regular expressions.

guide.bat (Windows) / guide.sh (Linux / Mac OS X) : The batch/bash-script we are going to program

(matthew|luke|mark|john).tok.low: The text files containing the four Gospels, already tokenized and lowercased

In order to complete our task, we will use the script guide.bat/guide.sh to do the calls.

Important SRILM PATH: Within the first lines of the script, you must change the SRILM path to the local path of your computed where you have the programs *ngram* and *ngram-count* of SRI Language Modeling Toolkit.

Furthermore you will see that the guide.bat / guide.sh file has some parts marked with "`REM to be programmed`" or "`# to be programmed`" that should be changed by your own code.

Making a robust 3-gram LM based on the Gospels

1. The first thing we are going to do is to build our **train** and **test** datasets:

   train : We are going to fuse the Gospels of Matthew, Luke and Mark. (That is done in the line 6 of your *guide* script and generates a new file named *bible.train.tok.low*.)

   test : the Gospel of John.

2. Once we have our train, and test data, we make the counts of the unigrams, bigrams and trigrams of the training data. That is done in the line number 10 of the *guide* script. (You don't have to do anything)

   - You can have a look at the files that have been generated. Open them with your editor and **answer the question 1 of the Assignment Quiz.**

3. Next, we are going to proceed to generate the Count of Counts and Good-Turing estimates before computing the Language Models. That is done in the line number 14 of your *guide* script.

   - You can **answer questions 2 and 3 of the Assignment Quiz.**

4. Given the following table template:

Counts (r)	**Count of Counts (N_i)**	r^*	**Coeff^0**	**Coeff^{Norm}**
1				
2				
3				
4				
5				
6				
7				
8	128	–	–	–

   (a) Fill the *Count of Counts* column with the data from trigrams of the former file. Row 5 has been already filled for you. **Put the results in question 4 of the Assignment Quiz.**

   (b) Compute the expected trigrams counts column (r^*) according to Good-Turing discounting method. **Put the results in question 5 of the Assignment Quiz (rounded to 4 decimals).**

(c) As we want to use the discounting as a weighting coefficient. Divide the r^* column by the *Counts* column. **Put the results in question 6 of the Assignment Quiz (rounded to 4 decimals).**

(d) We want to proceed to some normalization as we will only do discounting on the counts between 1 and 7. In order to do that we first compute the normalization coefficient based on the maximum slope between r_{min}=1 and r_{max}=7. That is:

$$\text{Common Term} = (\text{r}_{\text{max}} + 1) \cdot \frac{\text{N}_{\text{max+1}}}{\text{N}_{\text{min}}} \tag{1}$$

$$\text{Coeff}_i^{Norm} = \frac{\text{Coeff}_i^0 - \text{Common Term}}{1 - \text{Common Term}} \tag{2}$$

Fill the Coeff^{Norm} column according to this normalization. **Put the results in question 7 of the Assignment Quiz (rounded to 4 decimals)**

(e) In line 14 of the script, adding the flag –lm <output_file> generates the Good-Turing language model based on training data. Does the Coeff^{Norm} column match with the estimates file computed by your program? **Upload that file into the first upload of the Assignment File Evaluation interface.**

5. Now, let us compute the estimates to Kneser-Ney:

(a) What estimates do we need? **Answer question 8 of the Assignment Quiz.**

(b) Can you reformulate line 14 in order to obtain tri-gram Kneser-Ney estimates? (see `http://www.speech.sri.com/projects/srilm/manpages/ngram-count.1.html`) **Write it on line 18 of your program:**

- **Trick**: Think about changing GoodTuring word (gt) to Kneser-Ney (kn)

- **(Remember to delete the REM word at the beginning of the line! !)**

6. Let us focus on the following n-grams:

History	**Head Word**	Counts
they	came	21
they	said	31
they came	to	6
they came	into	2
they said	unto	13
they said	among	2

Answer question 9 of the Assignment Quiz

7. Now we can proceed to build the Language Models. We also want to build Language Models according to, *a)* Witten-bell, *b)* Unmodified Kneser-Ney, *c)* Modified Kneser-Ney. Replace line 22 of your script in order to obtain those language models and **upload the corresponding files to the file upload interface (fields 2 to 4)**

8. Open the Good-Turing Language Model. Do the probabilities match with the ones computed manually? (Recall the ARPA stores the probabilities in log-10 format). **Answer Question 10 of the Assignment Quiz**.

9. We now want to see how good our Language Models are. Replace lines 28, 31, 34, 37 and 40 of your script to compute the perplexity of each LM towards john.tok.low. **Answer Question 11 of the Assignment Quiz.**

10. As last step, we want to see how similar or different are the Gospels among them.

    (a) Select the best discounting method we found so far and build a language model for each Gospel. (Line 44 of your script)

    (b) Compute and give perplexities between all gospel-specific language models and the other Gospels (lines 47-87). Does it make sense to `http://en.wikipedia.org/wiki/Synoptic_gospels`? **Answer questions 12 to 14 of the Assignment Quiz.**

Congratulations! Not only you are now a Language Model expert but also a bible studies expert!