Assignment: Exploring Entropy and Language Modeling

Deadline: Jan 31 23:59 100 points

1. Entropy of a Text

In this experiment, you will determine the conditional entropy of the word distribution in a text given the previous word. To do this, you will first have to compute P(i,j)P(i, j), which is the probability that at any position in the text you will find the word i followed immediately by the word j, and P(j|i)P(j | i), which is the probability that if word i occurs in the text then word j will follow. Given these probabilities, the conditional entropy of the word distribution in a text given the previous word can then be computed as:

H(J|I) = - \sum_{i \in I,j \in J}P(i,j)\log_{2}P(j|i)H(J | I) = − i∈I,j∈J∑P(i, j) log2 P(j | i)

The perplexity is then computed simply as

PX(P(J|I)) = 2^{H(J|I)}PX(P(J | I)) = 2H(J | I)

Compute this conditional entropy and perplexity for the file

TEXTEN1.txt

This file has every word on a separate line. (Punctuation is considered a word, as in many other cases.) The i,j above will also span sentence boundaries, where i is the last word of one sentence and j is the first word of the following sentence (but obviously, there will be a fullstop at the end of most sentences).

Next, you will mess up the text and measure how this alters the conditional entropy. For every character in the text, mess it up with a likelihood of 10%. If a character is chosen to be messed up, map it into a randomly chosen character from the set of characters that appear in the text. Since there is some randomness to the outcome of the experiment, run the experiment 10 times, each time measuring the conditional entropy of the resulting text, and give the min, max, and average entropy from these experiments. Be sure to use srand to reset the random number generator seed each time you run it. Also, be sure each time you are messing up the original text, and not a previously messed up text. Do the same experiment for mess up likelihoods of 5%, 1%, .1%, .01%, and .001%.

Next, for every word in the text, mess it up with a likelihood of 10%. If a word is chosen to be messed up, map it into a randomly chosen word from the set of words that appear in the text. Again run the experiment 10 times, each time measuring the conditional entropy of the resulting text, and give the min, max, and average entropy from these experiments. Do the same experiment for mess up likelihoods of 5%, 1%, .1%, .01%, and .001%.

Now do exactly the same for the file

TEXTCZ1.txt

which contains a similar amount of text in an unknown language (*just FYI, that's Czech* [*])

Tabulate, graph and explain your results. Also try to explain the differences between the two languages. To substantiate your explanations, you might want to tabulate also the basic characteristics of the two texts, such as the word count, number of characters (total, per word), the frequency of the most frequent words, the number of words with frequency 1, etc.

Attach your source code commented in such a way that it is sufficient to read the comments to understand what you have done and how you have done it.

Now assume two languages, L_1L1 and L_2L2 do not share any vocabulary items, and that the conditional entropy as described above of a text T_1T1 in language L_1L1 is EE and that the conditional entropy of a text T_2T2 in language L_2L2 is also EE. Now make a new text by appending T_2T2 to the end of T_1T1. Will the conditional entropy of this new text be greater than, equal to, or less than EE? Explain (This is a paper-and-pencil exercise of course!)

2. Cross-Entropy and Language Modeling

This task will show you the importance of smoothing for language modeling, and in certain detail it lets you feel its effects.

First, you will have to prepare data: take the same texts as in the previous task, i.e.

TEXTEN1.txt and **TEXTCZ1.txt**

Prepare 3 datasets out of each: strip off the last 20,000 words and call them the **Test Data**, then take off the last 40,000 words from what remains, and call them the **Heldout Data**, and call the remaining data the **Training Data**.

Here comes the coding: extract word counts from the **training data** so that you are ready to compute unigram-, bigram- and trigram-based probabilities from them; compute also the uniform probability based on the vocabulary size. Remember (TT being the text size, and VV the vocabulary size, i.e. the number of types - different word forms found in the training text):

p_0(w_i) = 1 / Vp0(wi) = 1/V
p_1(w_i) = c_1(w_i) / Tp1(wi) = c1(wi)/T
p_2(w_i|w_{i-1}) = c2(w_{i-1},w_{i}) / c_1(w_{i-1})p2(wi∣wi−1) = c2(wi−1, wi)/c1(wi−1)
p_3(w_i|w_{i-2},w_{i-1}) = c3(w_{i-2},w_{i-1},w_{i}) / c2(w_{i-2},w_{i-1})p3(wi∣wi−2, wi−1) = c3(wi−2, wi−1, wi)/c2(wi−2, wi−1)

Be careful; remember how to handle correctly the beginning and end of the training data with respect to bigram and trigram counts.

Now compute the four smoothing parameters (i.e. "coefficients", "weights", "lambdas", "interpolation parameters" or whatever, for the trigram, bigram, unigram and uniform distributions) from the **heldout data** using the EM algorithm. (Then do the same using the **training data** again: what smoothing coefficients have you got? After answering this question, throw them away!) Remember, the smoothed model has the following form:

p_s(w_i|w_{i-2},w_{i-1}) = l_0p_0(w_i)+ l_1p_1(w_i)+ l_2p_2(w_i|w_{i-1}) + l_3p_3(w_i|w_{i-2},w_{i-1}),ps(wi∣wi−2,wi−1) = l0p0(wi) + l1p1(wi) + l2 p2(wi∣wi−1) + l3p3(wi∣wi−2,wi−1),

where

l_0 + l_1 + l_2 + l_3 = 1l0 + l1 + l2 + l3 = 1

And finally, compute the cross-entropy of the **test data** using your newly built, smoothed language model. Now tweak the smoothing parameters in the following way: add 10%, 20%, 30%, ..., 90%, 95% and 99% of the difference between the trigram smoothing parameter and 1.0 to its value, discounting at the same the remaining three parameters proportionally (remember, they have to sum up to 1.0!!). Then set the trigram smoothing parameter to 90%, 80%, 70%, ... 10%, 0% of its value, boosting proportionally the other three parameters, again to sum up to one. Compute the cross-entropy on the **test data** for all these 22 cases (original + 11 trigram parameter increase + 10 trigram smoothing parameter decrease). Tabulate, graph and explain what you have got. Also, try to explain the differences between the two languages based on similar statistics as in the Task No. 2, plus the "coverage" graph (defined as the percentage of words in the **test data** which have been seen in the **training data**).

Attach your source code commented in such a way that it is sufficient to read the comments to understand what you have done and how you have done it.

- If you want to see the accents correctly, select ISO Latin 2 coding (charset=iso-8859-2) for viewing, but your programs obviously will (should) work in any case (supposing they are 8-bit clean). For those linguistically minded & wishing to learn more about the language, look here. We will be using texts in this language often, albeit you will never be required to learn it.)

Turning in the Assignment

- Create a separate directory `assign` for your submission. Create a main web page called `index.html` or `index.htm` in that directory. Create as many other web pages as necessary. Put all the other necessary files (.ps and .pdf files, pictures, source code, ...) into the same directory and make relative links to them from your main or other linked web pages. If you use some "content creation" tools related to MSFT software please make sure the references use the correct case (matching uppercase/lowercase).

- Pack everything into a single `.tgz` file:
```
cd assign
tar -czvf ~/FirstName.LastName.assign.tgz ./*
```

e.g. for Jan Novák
`tar -czvf ~/Jan.Novak.assign.tgz ./*`

- Send the resulting file by e-mail (as an attachment) to
`hajic@ufal.mff.cuni.cz`
with the following subject line:
`Subject: FirstName.LastName NPFL067 Assignment`
e.g. for Jan Novák
`Subject: Jan.Novak NPFL067 Assignment`

Unix lab accounts

For MFF UK students, please see http://www.ms.mff.cuni.cz/labs/unix. For others, please visit http://www.ms.mff.cuni.cz/students/externisti.html.

Grades

- The grading table is available in SIS.
- The final grade (or pass/fail for PhD students) will be determined by both the final exam and your assignment results in a 50:50 ratio.

Exam

- The exam is written (not oral), with about 6 major questions and some subquestions. You will have 60 minutes to write down the answers.
- To get an idea of the type of exam questions, please see the questionaire for one of the previous year's final exam (Questionnaire).

Plagiarism

No plagiarism will be tolerated. The assignment is to be worked on your own; please respect it. If the instructor determines that there are substantial similarities exceeding the likelihood of such an event, he will call the two (or more) students to explain them and possibly to take an immediate test (or assignment, at the discretion of the instructor, not to exceed four hours of work) to determine the student's abilities related to the offending work. *All* cases of confirmed plagiarism will be reported to the Student Office.

Lateness

- For each day your submission is late, 5 points will be subtracted from the points awarded to the solution or a part of it, up to max. of 50 points per homework.
- Submissions received less then 4 weeks before the closing date of the term will not be graded and will be awarded 0 points.

Required Reading

Foundations of Statistical Natural Language Processing

Manning, C. D. and H. Schütze. *MIT Press*. 1999. ISBN 0-262-13360-1.

Eight copies of this book are available at the CS library for borrowing. Please be considerate to other students and do not keep the book(s) longer than absolutely necessary.

Recommended & Reference Readings

Speech and Language Processing

Jurafsky, D. and J. H. Martin. *Prentice-Hall*. 2000. ISBN 0-13-095069-6.

Three copies of Jurafsky's book are available at UFAL's library.

Programming PERL

Wall, L., Christiansen, T. and R. L. Schwartz. *O'Reilly*. 1996. ISBN 0-596-00027-8.

(Sorry no large cover picture available.)

Natural Language Understanding

Allen, J.. *Benajmins/Cummings Publishing Company* 1994. ISBN 0-8053-0334-0.

Elements of Information Theory

Cover, T. M. and J. A. Thomas. *Wiley*. 1991. ISBN 0-471-06259-6.

Statistical Language Learning

Charniak, E. *MIT Press*. 1996. ISBN 0-262-53141-0.

Statistical Methods for Speech Recognition

Jelinek, F. *MIT Press*. 1998. ISBN 0-262-10066-5.

Four copies of Jelinek's book are available at UFAL's library, but they are primarily reserved for those taking Nino Peterek's and/or Filip Jurcicek's courses.

Proceedings of major conferences (related to Natural Language Processing)

Some of the Proceedings are available at UFAL's library, physically and/or in electronic form. Most of them are, however, freely available through the ACL Anthology, including all volumes of the Computational Linguistics journal and the new Transactions of the ACL journal.

- ACL (Association of Computational Linguistics)
- European Chapter of the ACL
- North American Chapter of the ACL
- EMNLP (Empirical Methods in Natural Language Processing)
- COLING (International Committee of Computational Linguistics)
- ANLP (Applied Natural Language Processing, by ACL)
- ACL SIGDAT, other SIG (Special Interest Groups) Workhops, such as WVLC (Workshop on Very - Large Corpora)
- DARPA HLT (Defense Advanced Research Project Agency Human Language Technology Workshops)

Other Resources

- CLSP Workshops: Language Engineering for Students and Professionals Integrating Research - and Education
- Some interesting statistics on Czech and English.
- Eric Brill's short guide to Perl.
- Eric Brill's transformation-rules-based error-driven tagger (for Unix-based systems).

Malostranské náměstí 25
118 00 Praha
Czech Republic

+420 951 554 278 (phone)
ufal@ufal.mff.cuni.cz

find us on facebook
Page curated by hajic | Sign in

Powered by powered by drupal 7