CS838-1 Advanced NLP Homework 2

Due 2/27/2007 in class

Instructor: Jerry Zhu, jerryzhu@cs.wisc.edu

Type your answers and hand in a printed version to the instructor in class on the due date. The homework is worth 50% if it is no later than 48 hours (you may email me a pdf file), and worth nothing after that. I will not accept homeworks to the TA or in the physical mailbox.

Note: This homework requires running programs in Unix. Please talk to the TA for linux/unix help.

1 Simple Language Models

Let the training corpus be the sentence:

```
the quick brown fox jumps over the lazy dog
```

Let the vocabulary be all words above, plus the word `cat`.

Question 0. [20]

1. Use add-1 smoothing, create a unigram language model by hand. Write down the probability of each word in the vocabulary.
2. Compute the probability of the sentence

   ```
   the cat jumps over the dog
   ```

3. Use add-1 smoothing, create a bigram language model by hand (you do not need to create the full conditional probability table). Compute the probability of the above sentence. Show how you compute each conditional probability in the sentence (let $p(the|sentence\ begin) = p(the)$, the smoothed unigram probability).

2 Language Modeling with the CMU-Cambridge Toolkit

Download the CMU-Cambridge LM toolkit from `http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html`. Follow the documentation to `make install` it (check endian please). This should produce a set of executables in 'bin/'.

Download the training corpus from `http://www.cs.wisc.edu/~cs838-1/dataset/polarity-dataset-v2.0/training.text`. These are 1000 movie review articles. You will notice they have one sentence per line, and there is a sentence-beginning token `<s>` in front of each sentence. Do not further process the corpus. We will train language models on this corpus. Please follow the ML toolkit documentations in the following steps.

> **Question 1.** [**5**] Using `text2wfreq` and `wfreq2vocab`, create a vocabulary for words that *appear more than 4 times* (i.e., a count of at least 5). How many word types are there in your vocabulary?

With the above vocabulary, use `text2idngram` to collect unigram (specify the flag `-n 1`) counts from training.text. Create a `context cue` file `movie.ccs` with a single line `<s>` in it—We tell the program that this is the special sentence-beginning symbol. Use `idngram2lm` to create a *unigram* LM (use `-binary`, `-context` and `-n 1` flags). Save this unigram LM for later use.

> **Question 2.** [**10**] Using `evallm`, and the interactive command `perplexity`, compute the perplexity of the unigram LM on *test.text* (download from the same address above). What is the perplexity on test.text? What is the perplexity on the training corpus itself (training.text)?
>
> **Question 3.** [**5**] Repeat from `text2idngram`, but this time collect and build a *bigram* LM. What is the perplexity of the bigram LM on test.text and training.text?
>
> **Question 4.** [**5**] Collect and build a *trigram* LM. What is the perplexity of the trigram LM on test.text and training.text?
>
> **Question 5.** [**10**] Discuss the difference between test and training perplexity, as you move to more complicated LMs. Why training corpus perplexity is not a reliable measure of LM quality?

Now make a copy of your vocabulary file. Edit the copy:

- The first 4 lines starting with `##` are comments. Remove them so that the file has one word type per line.
- Remove `<s>` from the copy.

Run `evallm` again with the *unigram LM*. Run `perplexity` on the copy, this time with a `-probs vocab.probs` flag. The file *vocab.probs* contains the unigram probabilities of each word type, in the order specified in the copy.

> **Question 6.** [**5**] Find the unigram probability of the following words in vocab.probs:
>
> - the
> - movie
> - mulan
> - album

3 Random Sentence Contest

Call the distribution in vocab.probs p. Write a *sampling* program that samples words from p.

Question 7. **[10]** Sample 10,000 words from p. Write down the counts of the following words in your sample:

- the
- movie
- mulan
- album

Question 8. **[10]** We will have a contest on "the most interesting random sentence". From the random word sequence your sampler generated, pick a subsequence of words that you think are most interesting. Rules:

1. it must be a continuous subsequence;
2. there is no limit on length;
3. you may add or remove punctuations anyway you want—this is the only edit allowed.

Write down the sentence. Everyone who submit their sentence will get full score for this question. We will vote for the most interesting sentence in class, and the winner gets 15-minute fame.

4 Add-1 Smoothing as MAP Estimate

Question 9. **[10]** Prove that add-1 smoothing is the MAP estimate, with a Dirichlet prior with hyperparameters 2. Hint: formulate the problem as constrained optimization, and apply Lagrange multiplier.

5 KL-Divergence and MLE

Question 10. **[10]** Let $w_{1:n} \sim p$. Prove that finding the unigram MLE q on $w_{1:n}$ is equivalent to finding q that minimizes the KL-divergence $KL(p\|q)$, as $n \to \infty$.