Midterm Exam

Computational Linguistics (Fall 2014)

October 23, 2014

Your name: _____

This test is worth 60 points, and counts for 15% of your total grade. There is an extra credit section at the end which is undervalued. Attempt it only after completing the other questions.

You may refer to any books or notes that you have with you. Other than simple calculators, no electronic devices are allowed.

Write all your answers in the space provided. Use the reverse side of the pages if you need more room or scratch space. **You need not simplify arithmetic expressions in your solutions.** Show all your work. The exception is multiple choice questions.

Read the questions carefully and raise your hand for clarifications. The test is not ordered by difficulty, so if you find yourself getting stuck, move on and revisit the problem later.

Do not start reading or working on the exam until instructed.

Contents

1 Probability and N-Grams (20 points)

1. (3 points) A lipogram is a text where the letter `e` is avoided completely.

   (a) (1 point) The unigram probability of the letter `e` in English is 0.15. What the expected number of times that `e` appears in a text of length 10?

   (b) (1 point) What is the probability that a random text of length 10, generated by unigram letters, turns out to be a lipogram?

   (c) (1 point) You train a unigram letter model on a lipogram written in English. How do you think the entropy of this model compares to the entropy of a model trained on a standard (not necessarily lipogrammatical) English text?

   ☐ Lipogram entropy is smaller ☐ They're equal ☐ Lipogram entropy is greater

2. (1 point) The NSA creates a survey consisting of yes/no questions, to be handed out to all American residents, with the objective of uniquely characterizing every individual. What is the minimum number of questions needed on the survey?

3. (7 points) You see the word "`your`" written in isolation. Knowing that people commonly misspell "`you're`" as "`your`", hypothesize whether the writer actually meant to say "`your`", or intended to say "`you're`" and misspelt it. You're given this information:

   - The probability of meaning to say "`you're`" is twice the probability of "`your`".
   - 40% of people who mean to write "`you're`" misspell it as "`your`".
   - 10% of people who mean to write "`your`" misspell it (as something besides "`your`").

   Tip: Use distinct variable names for the events 'wrote your' and 'meant your' to avoid chaos.

(a) (4 points) Did the writer of "`your`" intend to say "`your`" or "`you're`"?

(b) (3 points) Consider a new scenario where the proportion of people who misspell "`your`" is 20% rather than 10%. Can you determine the writer's intention?

4. (1 point) I have a unigram word model trained on a *typical* standard English corpus. How would you expect the perplexity of the model on the sentence '`the sandwich Sally ate`' to compare to the perplexity of the same model on '`Sally ate the sandwich`'?

   ☐ Perplexity on '`the sandwich Sally ate`' is greater

   ☐ The perplexities are equal

   ☐ Perplexity on '`Sally ate the sandwich`' is greater

5. (1 point) What happens if I use a bigram word model instead?

   ☐ Perplexity on '`the sandwich Sally ate`' is greater

   ☐ The perplexities are equal

   ☐ Perplexity on '`Sally ate the sandwich`' is greater

6. (1 point) You have an n-gram distribution which you smooth using add-σ, for some $\sigma > 0$. The entropy of the smoothed distribution is

   ☐ smaller than ☐ equal to ☐ greater than

   the entropy of the original distribution.

7. (1 point) Compute $P(\texttt{ate}|\texttt{Sally})$ – i.e., the probability of the word `ate` given the previous word to the left is `Sally` – from these bigram counts.

`Sally sandwich`	10
`ate Sally`	5
`Sally the`	15
`Sally ate`	20
`ate the`	25
`sandwich ate`	1
`sandwich Sally`	2

8. (1 point) Give two examples of real-world systems that use n-grams.

9. (1 point) A language where the frequent words are shorter in length is more efficient than a language where the frequent words are long and rare words short. What do I mean by "efficient"?

   ☐ The entropy of the unigram word model is lower

☐ The total number of words in the vocabulary is smaller

☐ The average sentence length is lower

10. (1 point) The maximum likelihood estimate of the probability of an n-gram is its

    _____.

11. (2 points) Recall that, given a text w, finding an n-gram model θ with minimum perplexity $PPL(\theta, w)$ on the text is the same as finding the one maximizes the text's probability $P(w|\theta)$. Write a closed-form expression (no summation or product symbols) for $PPL(\theta, w)$ in terms of $P(w|\theta)$. Let $|w|$ denote the length of the text.

2 Model Design (20 points)

1. (11 points) Draw automata or write grammars for the following problems.

   (a) (2 points) Write a context-free grammar for a language where every string is of the form $a^n b^{n+m} c^m$ (some number n of `a`'s, followed by $m + n$ `b`'s, followed by m `c`'s; e.g. `abbbcc`). Hint: This is a small extension to the $a^n b^n$ grammar.

(b) (2 points) Draw a finite state acceptor that accepts noun phrases of the forms `Det Adj* Noun` or `Adj* Noun` (one or no determiner followed by zero or more adjectives followed by exactly one noun). The alphabet is {`Det`, `Adj`, `Noun`}.

(c) (4 points) Create a left-regular grammar of production rules for the above noun-phrase language. Recall that rules of left-regular grammars can only be of the form `A` $\rightarrow$ `Bc` or `A` $\rightarrow$ `c` or `A` $\rightarrow \varepsilon$, where `A` and `B` are non-terminals, and `c` is a terminal. Here, {`Det`, `Adj`, `Noun`} are terminals.

(d) (3 points) Draw a finite state transducer that converts English singulars to plurals. The alphabet is restricted to the phonemes {`k, g, i, s, z`}, and a word is input to the machine as a sequence of phonemes. If the word ends with:

- `s` or `z`, add `i z` to the end of the word. E.g. `kis` (the phonemic representation of "kiss") becomes `kisiz`

- `k`, add `s` to the end of the word. E.g. `kik` ("kick") becomes `kiks`
- `i` or `g`, add `z` to the end. E.g. `sig` becomes `sigz`

2. (6 points) Put on your creative hat and come up with a probabilistic generative story for the data given the tasks described below. Use words or diagrams. For example, our generative story for a misspelt word w, given the task of spelling correction, was

   - Generate a word with the correct spelling c according to an n-gram language model probability $P(c)$.
   - Transform c into a misspelling w according to a channel model probability $P(w|c)$ which is inversely proportional to the edit distance of w from c.

   (a) (3 points) A set of cognates – words in different languages that have a common origin; e.g. house (English), Haus (German), hus (Swedish). Task: reconstruct the word in the ancestor language that the cognates were derived from.

(b) (3 points) A speech recording of a sentence in German. Task: produce a transcription of the English translation of that sentence.

3. (2 points) Give an example of an English sentence that has two or more meanings caused by *structural* ambiguity (multiple parse trees with the same part-of-speech tags for all the words) rather than *lexical* ambiguity (multiple POS tags for some words). `I shot an elephant in my pajamas` is an example of a sentence with structural ambiguity, whereas `fruit flies like a banana` is not. No tree or explanation necessary. Only 1 point for PP-attachment examples that resemble the elephant-shooting one.

4. (1 point) What is your favorite topic in the course so far?

---

3 Hidden Markov Models (15 points)

1. (2 points) Let's look again at the sentence, `Sally ate the sandwich`. Our inventory consists of 30 part-of-speech tags, and we're given a bigram part-of-speech model.

   (a) (1 points) How many tag sequences are possible for the above sentence in total?

(b) (1 points) How many operations do you have to perform in the algorithm to find the most likely tag sequence for the above sentence?

2. (2 points) EM involves iterative applications of an expectation step and a maximization step. What, precisely, is the maximization step in the algorithm for finding maximum likelihood parameters of an HMM?

3. (3 points) Name the algorithm you would apply to each of the following problems:

   (a) Given the spelling of a word, a pronunciation-to-spelling model, and a trigram model of phoneme sequences, find the pronunciation of the word.

   _____

   (b) Given a corpus of speech clips and their transcriptions, find the parameters of the acoustic models (mappings from phonemes to sounds).

   _____

   (c) I'm part of the way through reading a novel. Under a part-of-speech bigram model for generating text, find the probability of the remaining contents of the novel.

   _____

4. (8 points) Sally only eats sandwiches. She picks the sandwich based on what meal of the day it is: **B**reakfast, **L**unch, or **D**inner. She never skips lunch, but she sometimes skips breakfast or dinner or both. However, she never skips two consecutive meals. (Dinner and breakfast are considered consecutive meals in this question.) For breakfast, she may eat **J**am toast or a **G**rilled cheese, for lunch, she chooses between a **G**rilled cheese and a **P**astrami, and for dinner, she always eats a **P**astrami. Based on a record of her sandwich purchases over several days, she would like to reconstruct which meals she has been eating.

(a) (4 points) Draw an HMM that illustrates the above scenario. Clearly denote the states, transitions, and emissions.

(b) (1 point) Which of the following sequences of purchases (spanning multiple days) are **not** possible? Select all that apply.

☐ JGGP ☐ GPPJ ☐ PJJG ☐ PPPP

(c) (3 points) Let $\alpha_i(S)$ denote the forward probability of a purchase record from the beginning up until time step i, ending with the state S, and $\beta_i(S)$ denote the backward probability of the record starting at state S, from i till the end, as defined in class. Given a purchase record of length 10, write the expression for the expected number of times that Sally ate breakfast in terms of α and β.

4 Vector Space Models (5 points)

1. (1 point) Which feature representation is the most appropriate for measuring syntactic similarities between words?

   ☐ Context vectors ☐ Term-document, unigram terms ☐ Term-document, n-gram terms

2. (1 point) The k-means algorithm for clustering alternates between

   computing, for each data point, _____

   and re-estimating _____

3. (3 points) Fill in the blanks in this pseudocode for learning a linear classifier.

   (a) Initialize the weight vector $\mathbf{w}$ to zero.

   (b) For each data point $\mathbf{x}_i$ and corresponding label y_i (where $y_i = +1$ or -1):

   - If _____ and _____,

     or _____ and _____, do nothing.

   - If _____ and $y_i = +1$, update $\mathbf{w} =$ _____

   - If _____ and $y_i = -1$, update $\mathbf{w} =$ _____

   (c) Repeat step (b) until there are no more mistakes.

5 Extra Credit (15 points)

Work on this section only after you're done with the previous questions.

1. (3 points) It is by arbitrary convention that n-gram models go left to right, treating the words to the left as the context. We can just as well flip the order and treat the words to the right as context instead, and compute the probability by reading the sentence right to left. Prove that the probability of a sentence w of length n under a left-ordered

bigram model θ is the same as under a right-ordered model ϕ; i.e.

$$P(w|\theta) = P(w_0|\texttt{<s>};\theta)P(\texttt{</s>}|w_n;\theta)\prod_{i=1}^{n} P(w_i|w_{i-1};\theta)$$

$$P(w|\phi) = P(\texttt{<s>}|w_0;\phi)P(w_n|\texttt{</s>};\phi)\prod_{i=0}^{n-1} P(w_i|w_{i+1};\phi)$$

$P(w|\theta) = P(w|\phi)$ when θ and ϕ are estimated from the same corpus.

2. (5 points) A *semiring* is a set together with two operations, $\oplus$ and $\otimes$. (If you know what a ring is, a semiring is simply a ring without the requirement of additive inverses.)

   - Additive and multiplicative associativity: $\forall a, b, c, (a \oplus b) \oplus c = a \oplus (b \oplus c)$ and $(a \otimes b) \otimes c = a \otimes (b \otimes c)$
   - Additive commutativity: $\forall a, b, a \oplus b = b \oplus a$
   - Distributivity: $\forall a, b, c, a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$
   - Additive and multiplicative identity: $a \oplus 0 = a$ and $a \otimes 1 = a$.
   - Annihilation: $a \otimes 0 = 0$

   We have been working on HMMs where the weights are probabilities, which forms a semiring called the *probability semiring*. This semiring consists of real numbers in $[0, 1]$, with $\oplus$ = ordinary addition and $\otimes$ = ordinary multiplication.

I claim that the Viterbi and forward algorithms are **exactly the same**, except that the weights that they work with are from different semirings. Let's say the weights for the forward algorithm (ignoring underflow issues) come from the probability semiring described above. Describe the semiring – the set, $\oplus$ operation, and $\otimes$ operation – that provides the weights for the Viterbi algorithm that would make the computation equivalent to the forward algorithm.

3. (7 points) Write a context-sensitive grammar to generate strings of the form $a^n b^n c^n$. The rules of a context-sensitive grammar are of the form

   $\alpha A \beta \rightarrow \alpha \gamma \beta$

   where A is a non-terminal, α and β are sequences of non-terminals and terminals, including ε, and γ is a sequence of terminals and non-terminals, not including ε. A rule is expanded in a tree only if the non-terminal A is flanked by α and β.