# Midterm Exam

Computational Linguistics (Fall 2013)

October 30, 2013

Your name: _____

This exam is worth 60 points, and counts for 15% of your total grade. There is a 10-point extra credit section at the end, which is severely undervalued, so attempt it only after completing the main exam.

Read the questions carefully. Raise your hand for clarifications.

Write all your answers in the space provided. Use the reverse side of the pages if you need more room or scratch space. You may refer to any books or notes that you have with you. Other than simple calculators, no electronic devices are allowed. **You need not simplify any arithmetic expressions in your solutions.**

The questions are not ordered by difficulty, so I recommend that if you find yourself getting stuck on a problem, work through the rest of the exam and revisit the problem later.

Show all your work. The exception is multiple choice questions: just select the answer.

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

## Contents

# 1 Probability and Information Theory (20 points)

1. (4 points) The NSA wants to create a survey consisting of a set of yes/no questions, to be handed out to all American residents. The survey is to be designed so that the answers will *uniquely determine the identity of every individual.* [1] Assume the survey is on paper, so they can't customize the number or choice of questions based on previous answers – everyone gets the same set of questions.

   (a) (2 points) The population of the country is 300 million. What is the minimum number of questions needed on the survey?

   (b) (2 points) Dartmouth has 6000 students. The NSA wants to save paper and make a shorter survey just for students at Dartmouth. What is the number of questions they *save* on the shorter survey compared to the ones given to all Americans?

2. (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

3. (9 points) You see the word `loose` written in isolation. Knowing that people commonly misspell `lose` as `loose`, can you hypothesize whether the writer actually meant to say `loose`, or intended to say `lose` but misspelt it? You're given this information:

[1] Most people have not studied information theory, so it easy to fool them into taking such a survey.

- The prior probability of the word `lose` is 3 times the prior probability of `loose`.
- 30% of people who mean to write `lose` misspell it as `loose`.
- 5% of people who mean to write `loose` misspell it (as something besides `loose`).

(a) (5 points) Did the writer of `loose` most likely intend to say `lose` or `loose`?

(b) (4 points) Consider the scenario where the proportion of people who misspell `loose` as something else is 10%, rather than 5%. Under this new scenario, can we determine if the writer of the word `loose` meant to say `lose` or `loose`?

4. (6 points) A lipogram is a text where the letter `e` is avoided completely.

   (a) (2 points) The unigram probability of the letter `e` in English is 0.2. What the expected number of times that `e` appears in a text of length 10?

   (b) (2 points) 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 are equal

   ☐ Lipogram entropy is greater

   ☐ Not enough enough information to answer

   (d) (1 point) Similarly, you train a unigram *word* model on an English lipogram. How do you think the entropy of this model compares to the entropy of a model trained on a standard English text?

   ☐ Lipogram entropy is smaller

   ☐ The entropies are equal

   ☐ Lipogram entropy is greater

   ☐ Not enough enough information to answer

# 2 N-Gram Language Models (10 points)

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

2. (1 point) I have a *bigram* word model trained on standard English. How would you expect the perplexity of the model on the sentence '`the sandwich Sally ate`' 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

3. (1 point) You have an n-gram distribution which you smooth using add-$\delta$, for some $\delta > 0$. The entropy of the smoothed distribution is

   ☐ smaller than ☐ equal to ☐ greater than

   the entropy of the original distribution.

4. (3 points) 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 |

5. (1 point) Some languages have vowel harmony, which is the constraint that all the vowels in a word must share a common feature (front, high, etc.). Assuming such

languages have few consonant clusters, what is the minimum order of n-grams over phonemes that can model the phonology?

☐ Unigrams ☐ Bigrams ☐ Trigrams ☐ 4-grams ☐ 5-grams

6. (2 points) N-gram models must be smoothed in order to prevent

   _____.

7. (1 point) The maximum likelihood estimate of an n-gram probability is its

   _____.

# 3 Finite State Design (10 points)

Draw finite state automata for the following problems. Denote all start and end states.

1. (5 points) A transducer that converts English singular forms to plurals. The highly simplified rules of going from singular to plural are:

   - If the word ends with `s`, add `e s` to the end of the word
   - In all other cases, add `s` to the end of the word

   Your alphabet is restricted to {`b, e, s`}, and words are input as character sequences. Draw a finite state transducer that convert singulars to plurals according to these rules.

2. (5 points) A machine that generates e-mail addresses. Symbols are restricted to the alphabetic characters `a` and `b`, the period `.` and the at-sign `@`. The address must contain exactly one at-sign and exactly one period in the domain name (the part that follows the at-sign). The username (the part before the at-sign), the first part of the domain name (preceding the period), and the second part of the domain name (following the period) must be at least one character long, and contain only alphabetic characters. Eg: `aab@ba.aa` and `abab@a.aba` are valid, whereas `aab.ba@aa.ab`, `@aa.ba`, `b@aa.` and `abab@ba.a.b` are invalid.

# 4 Hidden Markov Models (15 points)

1. (3 points) Let's look 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) (2 points) How many operations do you have to perform in the algorithm to find the most likely tag sequence for the sentence? Count each arithmetic evaluation

as a single operation; comparisons and bookkeeping are free.

2. (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. 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 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. You don't have to specify start or end states; assume that any of the states can be a start or end.

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

☐ JGGP ☐ GPPJ ☐ PJJG

(c) (3 points) Let $\alpha_i(S)$ denote the forward probability of a purchase record from first time step up until time step $i$, ending with the state $S$. $\beta_i(S)$ denotes the backward probability of the purchase record from time step $i+1$ onwards, starting from state $S$ at time step $i$.

Given a purchase record of length 10, write the expression for the expected number of times that Sally ate breakfast in that period, in terms of the $\alpha$ and $\beta$ values for the purchase record.[2]

3. (4 points) Name the algorithm you would apply to each of the following problems:

(a) Given a corpus of speech clips and their transcriptions, finding the parameters of the acoustic models (the models that map phonemes to sounds)

(b) Given the spelling of a word and a grapheme to phoneme model, finding the most likely pronunciation of the word.

[2]Hint: We've discussed how to compute the expected number of times Sally ate breakfast and a grilled cheese for breakfast. Just generalize.

(c) Finding the cognates of an English word in French

---

(d) I'm part of the way through reading a novel. Given a part of speech model for generating text, finding the probability of the remaining contents of the novel.

---

# 5 Context-Free Grammars (5 points)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | S | → | NP VP | 7 | VP | → | Verb NP |
| 2 | NP | → | Noun | 8 | Noun | → | sandwich |
| 3 | NP | → | the Noun | 9 | Noun | → | Sally |
| 4 | VP | → | VP and VP | 10 | Noun | → | president |
| 5 | VP | → | VP that S | 11 | Verb | → | saw |
| 6 | VP | → | Verb | 12 | Verb | → | ate |

For each sentence, determine if it can be generated by the above grammar.

1. Sally ate the sandwich and saw the president

   ☐ Yes ☐ No

2. the president saw that Sally ate sandwich

   ☐ Yes ☐ No

3. Sally saw the sandwich and the president ate the sandwich

   ☐ Yes ☐ No

4. the president that Sally saw ate

   ☐ Yes ☐ No

5. the sandwich saw that Sally ate that the president saw

   ☐ Yes ☐ No

**End of main exam. Continue for extra credit problems.**

# 6 Extra Credit (10 points)

Work on this section only after attempting the rest of the exam.

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 considering finite state machines where the weights are probabilities – namely, real numbers in $[0, 1]$, with ordinary addition and multiplication – which forms a semiring called the *probability semiring*. A more general form of weighted finite state machines draws weights from any semiring.

1. (1 point) In fact, what we have been calling unweighted finite state machines are actually weighted finite state machines where the weights come from the Boolean semiring consisting of the set $\{True, False\}$ ($True$ denotes the existence of an arc.) What are the $\oplus$ and $\otimes$ operations for a Boolean semiring?[3]

2. (4 points) 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 $\otimes$

[3]This should be an easy guess.

operation – that provides the weights for the Viterbi algorithm that would make the computation equivalent to the forward algorithm. [4]

3. (3 points) If we implement the forward algorithm in log space, describe the semiring that we are effectively using (ignoring underflow issues).

4. (2 points) Instead of probabilities, I can use non-negative integer costs for weights, and define the joint cost of an observation and state sequence as the sum of the costs of the transitions between the states, and emissions from states to symbols. In that case, my Viterbi algorithm should attempt to find the sequence of states with minimal joint cost (rather than maximal joint probability). Describe the semiring for this problem.

[4] Hint: recall that the Viterbi algorithm is usually implemented in log space.