CS4740/CS5740

Introduction to Natural Language Processing

Midterm Solutions
October 8, 2015

1 Some Basics (20 points)

1. (5 pts) Assuming that punctuation marks count as words, the number of word tokens is 17, and the number of word types is 13.

   Without treating punctuation as a word, the number of word tokens is 15, and the number of word types is 12.

   ## Grading guide

   - Assumptions clearly stated = 2pts. Not clear: -1pt. Not complete: -1pt.
   - Correct counts: both correct = 3pts; one correct: 2pts.

2. (5 pts) Answer can be TRUE or FALSE, depending on the explanation. Possibilities:

   (a) False. Trigram models make use of more context (longer history information, more precise), but due to data sparseness, it is extremely harder to train trigram models than bigram models.

   (b) True. Given enough training data, trigram models make use of more context / are more precise and/or more powerful.

   (c) False. Which order of n-gram will work the best depends on the data set in question and/or its size.

   ## Grading guide

   Partial score (2pts) for answers that discussed only the benefit of trigrams, without considering the practical downside of them.

3. (10 pts) I mistakenly made this question easier than I meant to by not making clear that I wanted 3 different **types** of ambiguities. Possible ambiguities:

   part of speech (POS) : “called” can be a past tense verb or a past participle.

   part of speech (POS) : “man” can be a noun or a verb.

WSD : “called” == called on the phone vs. called out to...other possibiities as well.

semantics or discourse : “James” refers to a particular person (in the world or in the document).

semantics or discourse : “Detroit” refers to a particular entity (in the world or in the document). item [syntax]: “from Detroit” can modify either “the man” or “called”.

semantics : James called from Detroit vs. the man was from Detroit.

Grading guide

4pts for the first correct ambiguity (3pts if unnamed or incorrectly named); plus 3pts(2pts) for the second; plus 3pts(2pts) for the third.

2 Lexical Semantics (20 points)

1. (10 pts) A lexeme is a 3-tuple of the orthographic form of a word (2pts), the pronunciation of a word (3pts), and a set of related meanings (3pts for the first; 2pts for the second).

   There can be: (a) one lexeme with at least two senses, or (b) two lexemes — one for the noun sense and one for the verb sense. E.g.:

   - walk (+2): [pronunciation of 'walk'] (+3): (the act of) traveling by foot (3pts); base on balls (baseball sense) (2pts); etc.

2. (2 pts) Homonyms are words that have the same spelling and pronunciation but have different (and unrelated) meanings.

   A well-worn example is the difference between a financial *bank* and a river *bank*.

Grading guide

1pt if the correct definition of homonymy is given, but no (correct) example.

3. (8 pts) Speech recognition systems conflate all of the lexemes of the homonym into one “group” — since the homonyms SOUND alike, they are treated in the same way in the speech system (4pts). Unfortunately, each element of the pair is of homonyms is likely to appear in very different contexts because they have very different meanings (4pts). So the words that (precede and) follow each element of the homonym pair will differ substantially (4pts). And the LMs used by the speech recognition system are likely to produce inappropriately high or low probabilities when predicting the next word (4pts).

Grading guide

See above. But a max of 8pts. -2pts for each **incorrect** statement.

3 WSD (15pts)

1. (5pts) No need to train (presumably on-the-fly) a separate classifier for each content word (5pts). Other answers possible, e.g. Ability (in theory) to pay attention to the entire context when disambiguating (4pts).

2. (5pts) Rather than directly evaluating a WSD system w.r.t. its ability to determine the correct word sense for a lexical item, an **extrinsic** evaluation determines the degree to which an embedded WSD system improves overall performance of the larger system. Lots of answers are possible:

   - machine translation
   - information extraction
   - question answering

   ## Grading guide

   For full credit, the answer should explain how the WSD contributes to the overall task. 2pts for a legitimate extrinsic task; 3pts for the explanation.

3. (5pts) For full credit, an answer should show at least two co-occurrence features for each of two senses. Co-occurrence features are words that are strongly associated with one (but not any other) senses for the target word.

   Examples for the "basketball/baseball/sports" sense: player, court, field, game, racket, catch, hit, ...

   Examples for the "dance" sense: dance, celebration, invitation, grand, gala, ...

   ## Grading guide

   3pts for features for one sense + 2pts for the second sense.

4 Language Modeling (25 points)

1. (10 pts) Maximum Likelihood Estimation

   Given that w_i is the ith word, we want is $P(w_i|w_{i-1})$, estimated as $\frac{count{-}of(t_i,t_{i-1})}{count{-}of(t_{i-1})}$

   $P(you|you) = \frac{0}{4}$ $P(do|you) = \frac{1}{4}$ $P(not|you) = \frac{0}{4}$ $P(like|you) = \frac{0}{4}$ $P(them|you) = \frac{0}{4}$ $P(.|you) = \frac{0}{4}$ $P(so|you) = \frac{0}{4}$ $P(say|you) = \frac{1}{4}$ $P(try|you) = \frac{0}{4}$ $P(!|you) = \frac{0}{4}$ $P(and|you) = \frac{0}{4}$ $P(may|you) = \frac{2}{4}$ $P(,|you) = \frac{0}{4}$ $P(i|you) = \frac{0}{4}$

   ## Grading guide

   4pts for the correct formulation of the problem (i.e. computing $P(x|you)$). 3pts for the (3) non-0 probabilities. 3pts for the unseen bigrams.

2. (5 pts) Laplacian smoothing

   $P(you|you) = \frac{1}{18}$ $P(do|you) = \frac{2}{18}$ $P(not|you) = \frac{1}{18}$ $P(like|you) = \frac{1}{18}$ $P(them|you) = \frac{1}{18}$ $P(.|you) = \frac{1}{18}$ $P(so|you) = \frac{1}{18}$ $P(say|you) = \frac{2}{18}$ $P(try|you) = \frac{1}{18}$ $P(!|you) = \frac{1}{18}$ $P(and|you) = \frac{1}{18}$ $P(may|you) = \frac{3}{18}$ $P(,|you) = \frac{1}{18}$ $P(i|you) = \frac{1}{18}$

Grading guide

3pts for the correct denominator. 2pts for the correct numerators. -2 if omitted unseen bigrams. -1 for small error in numerators or denominator.

3. (10 pts) Possible answers:

   - (Modify the training corpus to) **replace every first occurrence** of a (new) word type with an UNK token. Then treat UNK just like any other word token when gathering n-gram counts.
   - (Modify the training corpus to) **replace every word type that appears once** in the corpus with an UNK token. Then treat UNK just like any other word token when gathering n-gram counts.

Grading guide

Corpus modification = 8pts. Something about gathering counts for the new UNK word type = 2pts. -3pts if answer is really unclear (but it SEEMS as if the student had the right idea).