CS4740/CS5740
Introduction to Natural Language Processing
Midterm Solutions

1 Word Sense Disambiguation (15 pts)

Given the WordNet entries below, apply Lesk's (simple) dictionary-based word sense disambiguation algorithm to the target word **bass** in the following context '**bass** *playing maniac*'. Assume that stemming is applied during preprocessing of the test case (**bass** *playing maniac*) and WordNet entries. FOR FULL CREDIT, show the calculations step-by-step AND provide a brief description of each. E.g.
step 1: <description of step 1>
<calculations for step 1>
step 2: ...

bass
bass (the lowest portion of the musical range)
bass, basso (an adult male singer with the lowest voice)
sea bass, bass (the lean flesh of a saltwater fish of the family Serranidae)
bass (the member with the lowest range of a family of musical instruments)

playing
playing (the act of playing a musical instrument)
playing (the action of taking part in a game or sport or other recreation)

maniac
lunatic, madman, maniac (an insane person)
maniac (a person who has an obsession with or excessive enthusiasm for something)

Answer

1. consider the $bass^1$ sense and count the number of overlapping **content** words between the gloss definition of $bass^1$ with the glosses of **all** senses associated with **both** context words. **(1 point)**
   matches only *music*al for $playing^1$ **(1 point)**
   score of 1 **(1 point)**

2. do the same for *basso* sense **(1 + 1 points)**
   score of 0 **(1 point)**

3. do the same for *sea bass* sense **(1 + 1 points)**
   score of 0 **(1 point)**

4. do the same for *bass*^{4} sense **(1 point)**
   matches only *music*al and *instrument* of *playing*^{1} **(1 point)**
   score of 2 **(1 point)**

5. Return *bass*^{4}! **(3 points)**

Scoring: See points above.

2 Ambiguity in NLP (20 points)

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Figure 1: Final bottom-up chart.

1. (10 pts) We examined rather closely the Penn Treebank **part-of-speech (POS)** tag set shown above. It has been used extensively to annotate documents used to train machine-learning-based POS taggers in spite of the fact that some of its tags do not fully disambiguate the associated word tokens with respect to POS.

   Name one such problematic POS tag from the Penn Treebank tagset and **illustrate** this POS ambiguity **via an example sentence(s)**. I.e. show the relevant part-of-speech tags along with the words of the example sentence(s).

Answer: A correct answer needs to (1) specify a particular (single) POS tag, and (2) give an example of how a word tagged (correctly) with that POS tag can be used as two different parts of speech. For (2), we would typically be shown two different examples.

Case 1: TO. TO is used to denote the use of the word token "to" as either a preposition or to introduce an infinitival verb:

- I went to/TO the mall to/TO drive. (First use is as an infinitive; second, in an infinitive.)

Case 2: IN. IN is used to denote the use of a word token as either a preposition or to introduce a subordinate clause:

- Before/IN Claire reached him, Marseille had eaten all of her guacamole. (subordinate conjunction)
- Marseille ran for hours by/IN the river. (preposition)

Case 3: VBG. VBG is used to denote the use of a word token as either a progressive verb (ends in "ing") or as a gerund (noun ending in "ing"):

- Eating/VBG chocolate is very bad for dogs. (gerund)
- Marseille was running/VBG along the river when a deer appeared. (verb)

Scoring:

Correct problematic POS tag **(4 points)**. Only give **2 points** if multiple POS tags were given where one of them is correct. **-2 points** if one of the following POS tags was used (", ", (,), ., ;).

6 points total for the two examples (2pts each) and their explanation (1pt each).

2. (10 pts) Which of **homonymy** or **polysemy** would cause more problems for an n-gram-based language model of word prediction? Explain your answer.

   Answer: Homonyms would cause more problems. They have the same spelling but unrelated meanings (e.g. financial *bank* and river *bank*). For a language model, this means that there would be two very different contexts in which the homonym can occur and that would have to be captured via n-gram statistics. So the statistics associated with each use would likely be too sparse unless the corpus is very large.

   For polysemy (in which there are multiple **related** senses for a particular orthographic form), the contexts surrounding each sense are likely to be more similar (because their meanings are related) and so the n-grams associated with the associated word token can capture this shared context.

   Scoring:

5 points for "homonym" answer (or "it depends" answer if the argument is correct).
5 points for a correct argument.
If polysemy is given as the answer, then at most 5 points can be given if the explanation is reasonable, e.g. mentions different contexts for each meaning.

3 n-gram Language Models (30 points)

Mary had a little lamb , little lamb , little lamb . Mary had a little lamb . Its fleece was white as snow .

Assume that the above text is provided as the (entire) training corpus for a **bigram language model**. For preprocessing: assume that **all words are converted to lower case**; do not add beginning (or end) of sentence markers. No unknown word handling is required.

1. (10 pts) Using **Maximum Likelihood Estimation** and the **bigram model** derived from the above training data, compute *P(a little lamb was white)*.

   Answer + Scoring: P(a little lamb was white) =
   P(little | a) x P(lamb | little) x P(was | lamb) x P(white | was) = **(5 points)**
   2/2 x 4/4 x 0/4 x 1/1 = 0 (**5 points** for correct bigram probabilities and correct answer of 0.)
   It is also ok to include P(a) = 2/25 as the first term in the product.

2. (15 pts) Now using **add-one (Laplacian) smoothing** and the **bigram model** derived from the above training data, compute: *P(a little lamb was white)*.

   Answer + Scoring: (Important to know that the vocabulary size is 13.)
   P(a little lamb was white) =
   P(little | a) x P(lamb | little) x P(was | lamb) x P(white | was) = **(3 points)**
   (2+1)/(2+13) x (4+1)/(4+13) x (0+1)/(4+13) x (1+1)/(1+13)
   5 points for correct numerator calculations
   8 points for correct denominator calculations
   -2 for small errors.
   As above, it is ok to include P(a) in the product.

3. (5 pts) How many unseen bigrams are there for the *Mary had a little lamb* corpus?

   Answer + Scoring: The number of possible bigrams is $|V|^2 = 13x13 = 169$. **(2 points)**

   Of those, the number that appeared in the training text is 15. (mary-had, had-a, a-little, little-lamb, lamb-,, ,-little, lamb-., .-mary, .-its, its-fleece, fleece-was, was-white, white-as, as-snow, snow-.) **(2 points)**
   So the number of bigrams not encountered in the training set is: 169-15 = 154. **(1 point)**
   -1 for small errors.

4 HMMs (15 points)

Suppose you are doing Viterbi inference (i.e. applying the Viterbi algorithm) for a bigram HMM pos-tagger on this sentence: *All cows eat grass.* Suppose also that the algorithm had already progressed through the first two words (i.e. the program had "looked at" 'All' and 'cows'). Assume there are three parts of speech — N(oun), V(erb), D(eterminer). **What calculations will occur for the next word?** For your answer, please use the following notation:

- the words of the sentence are w_1, w_2, w_3, w_4;
- the corresponding tags are t_1, t_2, t_3, t_4;
- the scores are $s_{something}(w_i)$
- and the backpointers (or chains) are $b_{something}(w_i)$.

Just as a hint, you should end up with 3 scores and 3 backpointers (or chains) for the word 'eat'.

Answer:

$$s_N(w_3) = \max \left\{ \begin{array}{l} s_N(w_2) \times P(N|N) \\ s_V(w_2) \times P(N|V) \\ s_D(w_2) \times P(N|D) \end{array} \right\} \times P(eat|N)$$

$$b_N(w_3) = \cancel{\text{argmax index}} \quad \operatorname*{argmax}_{t \in \{N,V,D\}} s_N(w_3)$$

$$s_V(w_3) = \max \left\{ \begin{array}{l} s_N(w_2) \times P(V|N) \\ s_V(w_2) \times P(V|V) \\ s_D(w_2) \times P(V|D) \end{array} \right\} \times P(eat|V)$$

$$b_V(w_3) = \operatorname*{argmax}_{t \in \{N,V,D\}} s_V(w_3)$$

$$s_D(w_3) = \max \left\{ \begin{array}{l} s_N(w_2) \times P(D|N) \\ s_V(w_2) \times P(D|V) \\ s_D(w_2) \times P(D|D) \end{array} \right\} \times P(eat|D)$$

$$b_D(w_3) = \operatorname*{argmax}_{t \in \{N,V,D\}} s_D(w_3)$$

Figure 2: Viterbi Calculations

Scoring:

10 points for the correct s scores: $s_N(w_3)$, $s_V(w_3)$, $s_D(w_3)$.
5 points for the correct b scores: $b_N(w_3)$, $b_V(w_3)$, $b_D(w_3)$