Final Exam
Natural Language Processing

8th of September 2015

Remember to fill in your name on all pages
Good luck

Problem 1 (1 point). Compute the Good-Turing smoothing of the words *perch*, *trout* and *bass* in a corpus that contains only the following words (words are given with their counts in the table below):

carp	*perch*	*whitefish*	*trout*	*salmon*	*eel*
10	3	2	1	1	1

Problem 2 (2 points). Suppose we have the following short movie reviews, each labeled with a genre, either **comedy** or **action**.
1. *fun, couple, love, love* (**comedy**)
2. *fast, furious, shoot* (**action**)
3. *couple, fly, fast, fun, fun* (**comedy**)
4. *furious, shoot, shoot, fun* (**action**)
5. *fly, fast, shoot, love* (**action**)

Using a simple Naïve Bayes approach with Laplace smoothing, how would we classify the following review: *fun, couple, shoot, action*?

Problem 3 (3 points). In a set of 806,791 documents, we get the following data on a few terms and a few documents:

term	document frequency	Doc 1	Doc 2	Doc 3
car	18,165	27	4	24
auto	6,723	3	33	0
insurance	19,241	0	39	29
best	25,235	14	9	17

Compute the tf-idf value for these terms and documents. What is the cosine similarity between query "best car best insurance" and Doc 1, 2 and 3, respectively (use tf-idf weighting - nnc.ltc variation). Which of these three documents would be ranked first by a search engine using the nnc.ltc scheme?

Problem 4 (1 point). What is the mean average precision (MAP) for the following sequence of retrieved documents, where R denotes a relevant document and N denotes an irrelevant document? (Assume there are 20 relevant documents in the collection)

N R R N R R N R N R N R N R N

Problem 5 (3 points). Simulate the run of the Earley Parser for grammar G on the sentence *The large can can hold the water*, where

$G = (\{\text{S, NP, VP, D, J, N, V}\}, \{the, large, can, hold, water\}, \text{S}, R)$,

and the set R of rules is given by

$R = \{S \to \text{NP VP, NP} \to \text{D J N} \mid \text{D N} \mid \text{J N, VP} \to \text{V VP} \mid \text{V NP}\} \cup$
$\{\text{D} \to the, \text{J} \to large, \text{N} \to can|water, \text{V} \to can|hold\}$

Does the grammar G accept this sentence?

DEFINITION 1. Let w be a word that appears c times in the corpus, N_c = the count of things we've seen c times, and N = the total number of tokens in the corpus. Then,

$$c^*(w) = \begin{cases} \frac{(c+1)N_{c+1}}{N_c}, & \text{if } c > 0 \\ N_1, & \text{if } c = 0 \end{cases}$$

$$P^*_{GT}(w) = \frac{c^*(w)}{N}$$

DEFINITION 2. Let S be a sentence, C a class of possible labels and V the vocabulary. Then,

$$c_{NB}(S) = argmax_{c \in C} P(c) \prod_{w \in S} P(w \mid c),$$

where the prior probability of the class is $P(c) = doccount(c)/N_{doc}$, and the probability of a word w given the class c, computed using Laplace (add-1) smoothing with unknown words, is:

$$P(w \mid c) = \frac{count(w,c) + 1}{\sum_{v \in V} count(v,c) + |V| + 1}$$

DEFINITION 3. SMART Notation: denotes the combination in use in an engine, with the notation ddd.qqq, using the acronyms from the following table:

Term frequency	Document frequency	Normalization
n (natural): $tf_{t,d}$ l (logarithm): $1 + \log_{10}(tf_{t,d})$	n (no): 1 t (idf): $\log_{10}(\frac{N}{df_t})$	n (none): 1 c (cosine): $\frac{1}{\sqrt{\sum_i x_i^2}}$

DEFINITION 4. The Mean Average Precision (MAP) is the average of the precision value obtained for the top k documents, each time a relevant document is retrieved.

Precision = TP/(TP+FP)

TP = true positives (the number of gold answers that were correctly guessed)

FP = false positives (the number of guessed answers that were incorrect)

FN = false negative (the number of gold answers that were not correctly guessed)

DEFINITION 5. Start by building a sequence of state sets called *Earley sets*

Input: $x_1 x_2 \ldots x_n$

Sequence of state sets: $S_0, S_1, \ldots, S_n$

A set S_i contains various "tokens": $[A \to \alpha \bullet \beta, j]$

- *Scan*: $[A \to \ldots \bullet a \ldots, j] \in S_i$, $a = x_{i+1}$,
  add $[A \to \ldots a \bullet \ldots, j]$ to S_{i+1}
- *Predict*: $[A \to \ldots \bullet B \ldots, j] \in S_i$
  add $[B \to \bullet \alpha, i]$ to S_i for all rules $B \to \alpha$
- *Complete*: $[A \to \ldots \bullet, j] \in S_i$
  add $[B \to \ldots A \bullet \ldots, k]$ to S_i for all tokens $[B \to \ldots \bullet A \ldots, k]$ in S_j

Accept $x_1 x_2 \ldots x_n$ if in S_n there exist a rule of type $[S \to \alpha \bullet, 0]$