School of Computing and Information Systems
The University of Melbourne
COMP90042
WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2018)

Sample solutions for discussion exercises: Week 5

Discussion

1. What is the difference between supervised and unsupervised HMMs? How do you train each version?

   - Supervised HMMs assume you have a corpus annotated with tags and use that information to train the HMM. Unsupervised HMMs assume you only have observations (words), without any information about the latent variables (tags). Supervised HMMs are trained using MLE on the tagged corpus, through frequency counting, unsupervised HMMs use expected counts instead, and update the parameters using an iterative algorithm such as EM.

2. Can you give examples where you could apply unsupervised POS tagging?

   - Transferring a tagger to a different domain (News -¿ Twitter, for example)
   - Learning a tagger for a new language without tagged corpora.
   - When you have no tagged corpora for a specific domain but large amounts of unlabelled text (Twitter).

3. What is **chart parsing**? Why is it important?

   - **Parsing** in general is the process of identifying the structure(s) of a sentence, according to a grammar of the language.
   - In general, the search space is too large to do this efficiently, so we use a dynamic programming method to keep track of partial solutions. The data structure we use for this is a **chart**, where entries in the chart correspond to partial parses (licensed structures) for various spans (sequences of tokens) within the sentence.

4. Consider the following simple **context–free grammar**:

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

   (a) What changes need to be made to the grammar to make it suitable for **CYK parsing**?

      - For CYK parsing, a grammar needs to be written in Chomsky Normal Form, where each rule consists of either:

  - a (single) non-terminal which re-writes as a single terminal, or
  - a (single) non-terminal which re-writes as exactly two non-terminals
- Here, we have two rules where a non-terminal re-writes as three non-terminals (`VP -> V NP PP` and `NP -> Det N PP`); we remove these rules and replace them with the following:

```
VP -> V X
X -> NP PP
NP -> Det Y
Y -> N PP
```

(b) Using the CYK strategy and the above grammar in CNF, parse the following sentences:

i. "a man saw John"

ii. "an park by Bob walked an park with Bob"

iii. "park by the cat with my telescope"

- This sentence has no parse, because the cell [0,7] doesn't have an `S`.

a	*man*	*saw*	*John*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] S
	[1,2] N	[1,3] -	[1,4] -
		[2,3] V	[2,4] VP
			[3,4] NP

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] NP, X	[0,5] -	[0,6] -	[0,7] S	[0,8] -	[0,9] S, S
	[1,2] N	[1,3] -	[1,4] Y	[1,5] -	[1,6] -	[1,7] -	[1,8] -	[1,9] -
		[2,3] P	[2,4] PP	[2,5] -	[2,6] -	[2,7] -	[2,8] -	[2,9] -
			[3,4] NP	[3,5] -	[3,6] -	[3,7] S	[3,8] -	[3,9] S, S
				[4,5] V	[4,6] -	[4,7] VP	[4,8] -	[4,9] VP, VP
					[5,6] Det	[5,7] NP	[5,8] -	[5,9] NP, X
						[6,7] N	[6,8] -	[6,9] Y
							[7,8] P	[7,9] PP
								[8,9] NP

park	*by*	*the*	*cat*	*with*	*my*	*telescope*
[0,1] N	[0,2] -	[0,3] -	[0,4] Y	[0,5] -	[0,6] -	[0,7] Y
	[1,2] P	[1,3] -	[1,4] PP	[1,5] -	[1,6] -	[1,7] PP
		[2,3] Det	[2,4] NP	[2,5] -	[2,6] -	[2,7] NP, X
			[3,4] N	[3,5] -	[3,6] -	[3,7] Y
				[4,5] P	[4,6] -	[4,7] PP
					[5,6] Det	[5,7] NP
						[6,7] N