Tutorial 1: Regular Expressions and Text Normalization

Q1. Write regular expressions for the following languages. By "word", we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

1. The set of all alphabetic strings;
2. The set of all lower case alphabetic strings ending with a letter b;
3. The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug");
4. All strings that start at the beginning of the line with an integer and that end at the end of the line with a word;
5. All strings that have both the word "grotto" and the word "raven" in them (but not, e.g., words like "grottos" that merely contain the word "grotto");

HINT: Not all notions are covered in lectures, and it is fine that your RE cannot fully satisfy the specified requirements.

Q2. Try all your answers on http://regexr.com/ You may need to change the textbox to test two cases: the textbox contains one or more matched strings, and the textbox does not contain any matched string. What are the errors (e.g., false positive and false negative) have you observed?

Q3. Select all strings that can be matched by regular expression /E*F+[^Gg]/

A. EFG B. EF C. FFF D. EFFa

Q4. Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.

Q5. Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 2) of two sentences "computed the edit distance" to "the edit distance is computed". Show your work and show the alignment between the two strings. You may use edit distance defined at **word level** instead of character level.

Q1) 1 \b[a-zA-Z]+\b

Q2)

2 \b[a-z]*b\b

3 (\b[a-zA-Z]+\b)\s+\1

The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug"):

- (\b[a-zA-Z]+\b)\s+\1

Explanation

- [a-zA-Z]+ → all alphabetic strings
- \s → whitespace (space, tab..)
- \1 → used to refer back to the first pattern in the expression which is put **inside a parentheses** ( )
  - We may have \2 or \3 to refer to the second and third patterns put inside parentheses.

4 (\b[0-9]*[a-zA-Z]\b) ✗

All strings that start at the beginning of the line with an integer and that end at the end of the line with a word

- ^\d+\b.*\b[a-zA-Z]+$

Explanation

- \d → a digit
- \b → a word boundary
- ^,$ → the **beginning** and **end** of a line
- . → a wildcard expression that matches any single character (except a carriage return)
- * → Kleene star, zero or more occurrences of the immediate previous character or regular expression
- .* → any string of characters

5 (\b[a-zA-Z]*+(grotto|raven)+[a-zA-Z]\b) ✗

All strings that have both the word `grotto` and the word `raven` in them (but not, e.g., words like `grottos` that merely contain the word `grotto`)

- (.*\bgrotto\b.*\braven\b.*)|(.*\braven\b.*\bgrotto\b.*)

Explanation

- The two words grotto and raven may appear in any order.
- There could be other strings around the two words

Q3)

Expression

/E*F+[^Gg]/g

Text | Tests NEW

A. EFG → B. EF → C. FFF → D. EFFa

A. EFG → B. FFF → C. EFFa → D. EF

A. EFG → B. EF → C. EFFa → D. FFF

A. EFG → B. FFF → C. EFFa → D. EF

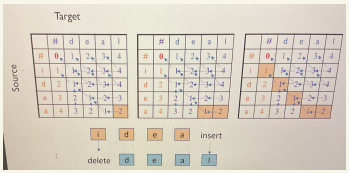
4)

		i	d	e	a
	0	1	2	3	4
d	1	1	1	2	3
e	2	2	2	1	2
a	3	3	3	2	1
l	4	4	4	3	2

ok row column? No need arrow?

(1) dea → delete i → dea(1)
insert l

Show this?



5)

		the	edit	distance	is	Computed
	0	1	2	3	4	5
Computed	1	2	3	4	5	4
the	2	1	2	3	4	5
edit	3	2	1	2	3	4
distance	4	3	2	1	2	3

Computed the edit distance → insert
insert ↓
delete the edit distance is Computed

Q1)
1) \b[a-zA-Z]+\b ✓
2) \b[a-z]*b\b ✓

Q2)
3) (\b[a-zA-Z]+\s+\1
4) ^[0-9]\w*[a-zA-Z]+$
5) \b\w*([grotto & raven]\w+[grotto & raven])\b

Q3) B, C, D

Q4)

	#	d	e	a	L
#	0	1	2	3	4
i	1	2	3	4	5
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2

i d e a *
| | | | |
* d e a L

1) remove i
2) insert L

Q5)

	#	the	edit	distance	is	Computed
#	0	1	2	3	4	5
Computed	1	2	3	4	5	4
the	2	1	2	3	4	5
edit	3	2	1	2	3	4
distance	4	3	2	1	2	3

Computed the edit distance * * Computed
| | | | | |
* the edit distance is

1) delete Computed
2) insert is
3) insert Computed

Tutorial 2: Text Normalization

Q1. Consider the following word segmentation algorithm in the lecture notes:

Given a lexicon of Chinese, and a string

1) Start a pointer at the beginning of the string

2) Find the longest word in dictionary that matches the string starting at pointer

3) Move the pointer over the word in string

4) Goto2

Strictly following the algorithm, you perhaps end up with failing to segment a string, if you cannot find a matching. For example, consider segmenting the following string using the given lexicon.

String: thetablesdownthere

Lexicon: the table down there bled own.

Discuss how to fix the above problem.

if Lexicon the table down there bled own theta

then string will be theta b l e s down there

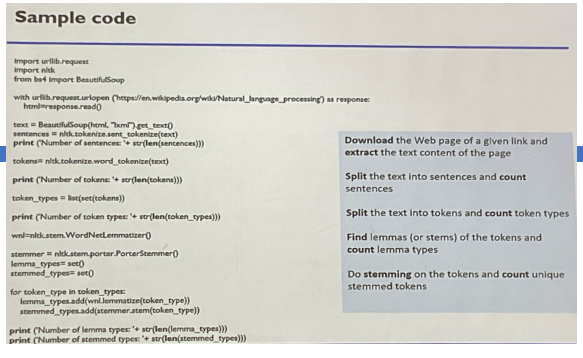
Q2. Try the tokenization demo on https://textanalysisonline.com/ (or you may use other tokenizer APIs, e.g., http://text-processing.com/demo/tokenize/). Discuss your findings based on the output of different tokenizers.

Q3. Try the stemmer demo on https://textanalysisonline.com/ (or you may use other stemmer APIs, e.g., http://text-processing.com/demo/stem/). Discuss your findings based on the output of the stemmers.

Q4. Write a program to do the following tasks:

1. Download the Web page of a given link and extract the text content of the page
2. Split the text into sentences and count sentences
3. Split the text into tokens and count token types
4. Find lemmas (or stems) of the tokens and count lemma types
5. Do stemming on the tokens and count unique 'stemmed' tokens

You may use any tools, including nltk, LingPipe, and Stanford NLP software.



Q1) ① <s> the tables downthere </s>

② <s> the tables down there </s>

TF is this TUT?

Q5. **Open-ended Question, for discussion only**.

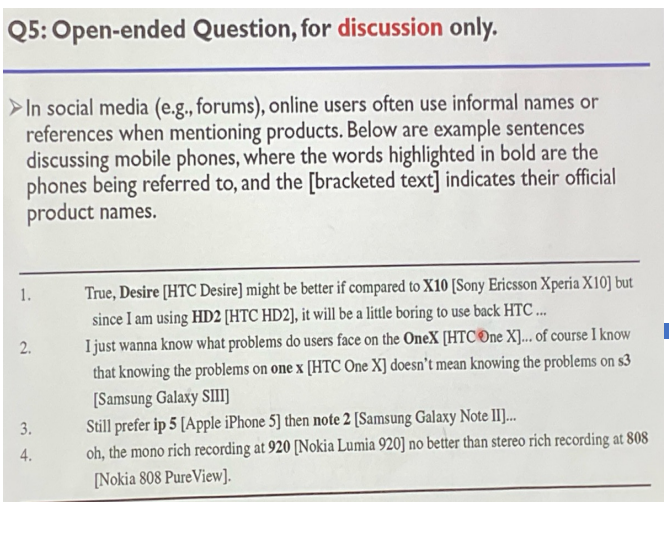
In social media (e.g., forums), online users often use informal names or references when mentioning products. Below are example sentences discussing mobile phones, where the words highlighted in bold are the phones being referred to, and the [bracketed text] indicates their official product names.

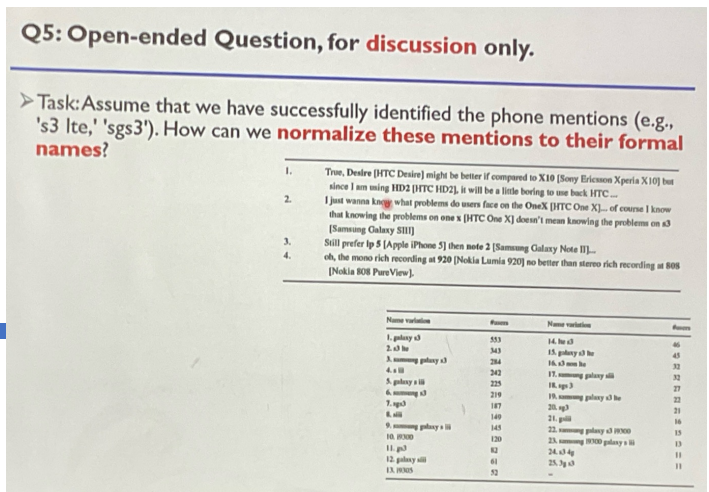
1. True, **Desire** [HTC Desire] might be better if compared to **X10** [Sony Ericsson Xperia X10] but since I am using **HD2** [HTC HD2], it will be a little boring to use back HTC ...
2. I just wanna know what problems do users face on the **OneX** [HTC One X]... of course I know that knowing the problems on **one x** [HTC One X] doesn't mean knowing the problems on **s3** [Samsung Galaxy SIII]
3. Still prefer **ip 5** [Apple iPhone 5] then **note 2** [Samsung Galaxy Note II]...
4. oh, the mono rich recording at **920** [Nokia Lumia 920] no better than stereo rich recording at **808** [Nokia 808 PureView].

The table below shows the number of users who have mentioned a phone using a specific name in a forum.

Name variation	#users	Name variation	#users
1. galaxy s3	553	14. lte s3	46
2. s3 lte	343	15. galaxy s3 lte	45
3. samsung galaxy s3	284	16. s3 non lte	32
4. s iii	242	17. samsung galaxy siii	32
5. galaxy s iii	225	18. sgs 3	27
6. samsung s3	219	19. samsung galaxy s3 lte	22
7. sgs3	187	20. sg3	21
8. siii	149	21. gsiii	16
9. samsung galaxy s iii	145	22. samsung galaxy s3 i9300	15
10. i9300	120	23. samsung i9300 galaxy s iii	13
11. gs3	82	24. s3 4g	11
12. galaxy siii	61	25. 3g s3	11
13. i9305	52	–	

Task: Assume that we have successfully identified the phone mentions (e.g., 's3 lte,' 'sgs3'). How can we normalize these mentions to their formal names?





Tutorial 3: N-gram and Language Model

always put sentence boundary (unless the ques state no need)

Q1. Given the following three word sequences (i.e., the corpus).

very good tennis player in US Open

tennis player US Open

tennis player qualify play US Open

(i) Build a table of bigram counts from the word sequences.
(ii) Compute the bigram probabilities using Laplace smoothing.

Q2. Write out the equation for trigram probability estimation, and use the equation to compute the trigram probability for $P(US|\ tennis\ player)$ and $P(player\ |\ good\ tennis)$ according to the corpus given in Q1.

Q3. Given the bigram probability in the following table, compute the probability of "I eat Chinese food". Explain how you compute the probability. State your assumptions and if more probability values are needed, you may use random values.

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Q4. Why do we need to do smoothing for language model?

Q5. Given some text, what are the general steps to collect all counts needed for building an n-gram language model?

connect

Q6. **For discussion only**: You are given a text collection of 100GB, and asked to train a bigram language model. You have a computer with 16GB ram and 1TB storage. Think about the best choices (steps) for implementation.

Q1

Use Laplace smoothing

10

w_{n-1} \ w_n	very	good	tennis	player	in	US	open	qualified	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
US	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualified	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1} \ w_n	very	good	tennis	player	in	US	open	qualified	play	</s>
<s>	2/13	1/13	3/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
very	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
good	1/11	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
tennis	1/13	1/13	1/13	4/13	1/13	1/13	1/13	1/13	1/13	1/13
player	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11
in	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
US	1/13	1/13	1/13	1/13	1/13	1/13	4/13	1/13	1/13	1/13
open	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	4/13
qualified	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	2/11	1/11
play	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11

<s>	very	good	tennis	player	in	US	open	qualified	play	</s>
3	1	1	2	3	1	3	2	1	1	3

Unique = 5 × 10

w_{n-1} \ w_n	very	good	tennis	player	in	US	open	qualified	play	</s>
<s>	$\frac{1+1}{3+5}=\frac{1}{4}$	1/8	3/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
very	1/8	$\frac{2}{8}=\frac{1}{4}$	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8
good	1/6	1/6	1/3	1/6	1/6	1/6	1/6	1/6	1/6	1/6
tennis	1/8	1/8	1/8	$\frac{4}{8}=\frac{1}{2}$	1/8	1/8	1/8	1/8	1/8	1/8
player	1/8	1/8	1/8	1/8	1/4	1/4	1/8	1/4	1/8	1/8
in	1/6	1/6	1/6	1/6	1/6	1/3	1/6	1/6	1/6	1/6
US	1/8	1/8	1/8	1/8	1/8	1/8	1/2	1/8	1/8	1/8
open	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/4	1/8	1/2
qualified	1/6	1/6	1/6	1/6	1/6	1/6	1/6	1/6	1/3	1/6
play	1/6	1/6	1/6	1/6	1/6	1/3	1/6	1/6	1/6	1/6

Q2

$$P(US \mid tennis\ player) = \frac{C(tennis\ player\ US)}{C(tennis\ player)} = \frac{1}{3}$$

$$P(player \mid good\ tennis) = \frac{C(good\ tennis\ player)}{C(good\ tennis)} = \frac{1}{1} = 1$$

Q3) Consider Sentece boundary

$P(\langle s \rangle \mid \text{eat Chinese food } \langle /s \rangle)$

$= P(\langle s \rangle) \times P(I \mid \langle s \rangle) \times P(\text{eat} \mid I) \times P(\text{Chinese} \mid \text{eat})$

$\times P(\text{food} \mid \text{Chinese}) \times P(\langle /s \rangle \mid \text{food})$

$= \quad ? \times ? \times 0.0036 \times 0.021$

$\times 0.52 \times ?$

Q4) to handles Out of Vocabulary words and avoid assigning zero probabilities to unseen event

Q5)

Tutorial 4: POS tagging and HMM

Q1. Find tagging errors in each of the following sentences that are tagged with the Penn Treebank tagset. You may get help from online demos of POS tagging services.

1. How/WRB do/MD I/PRP get/VB to/TO Singapore/NN
2. Do/VBP you/PRP have/VB any/DT vacancies/NN
3. This/DT room/NN is/VBZ too/JJ noisy/JJ
4. Can/VB you/PRP give/VB me/PRP another/DT room/NN

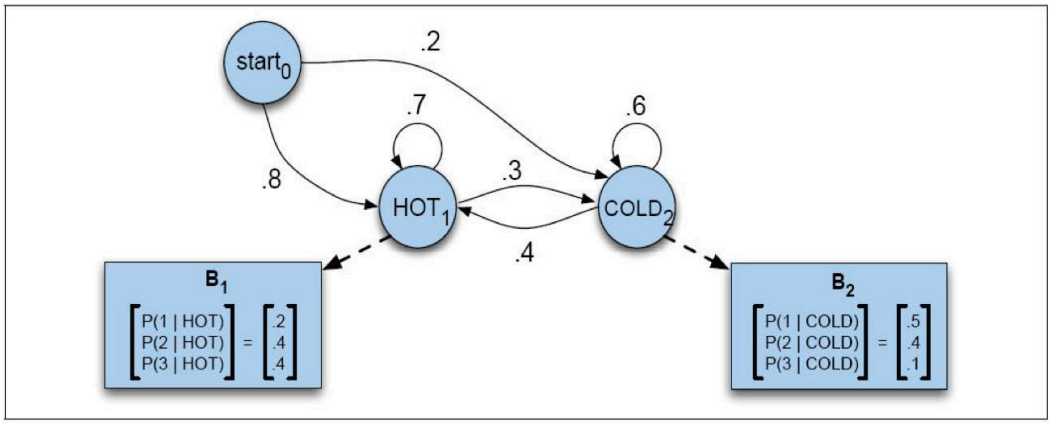
Q2. Compute the best tag sequence for "I want to race" using the Viterbi algorithm with the provided HMM parameters, i.e., the transition probability and the word likelihood probabilities.

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Q3. Run the Viterbi algorithm with the HMM in the figure below to compute the most likely weather sequences for each of the two following observation sequences. Note: You may consider using HMM packages for computation, e.g., https://pypi.org/project/hmmlearn/

- Sequence: 312312312
- Sequence: 311233112



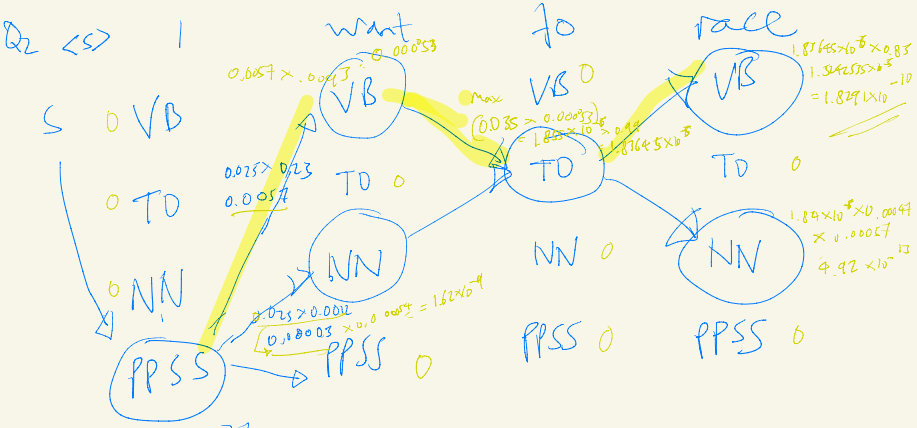
Q1) 1) How / WRB ✓
do / MD
I / PRP ✓
get / VB ✓
to / TO ✓
Singapore / NN ✗ → NNP

2) do / VBP ~~→ NNP~~ ✓
you / PRP ✓
have / VB ~~✗~~ ✓
any / DT ✓
vaccines / NN ✗ → NNS

3) too / RB

4) can / MD

Q2 <s> I want to race



Q4. The task of **negation scope detection** is to extract the parts of a sentence that is being negated. For example, in the sentence "I have not submitted my assignment", the negation scope is "submitted my assignment".

Formulate this problem as a sequence labelling task, and discuss how to apply Hidden Markov Model (HMM) to solve this problem. Clearly state the probabilities that need to be learned by the HMM.

Q1) TUT1
1) \b[a-zA-Z]+\b
2) \b[a-z]*b\b
3) (\b[a-zA-Z]{2}\b)\S+\1
4) ^[a-qs]*.*[^a-zA-Z]+$
5) .*([grotto].*[craven]) | ([craven].*[grotto]).*
(.*\bgrotto\b.*\bcraven\b.*) | (.*\bcraven\b.*\bgrotto\b.*)

Q3) B C D

Q4)

		d	e	a	l
	0	1	2	3	4
i	1	sub 1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2

i d e a *
* d e a L
- delete i
- insert L

TUT2

Q1

TUT3

Q1) <s> very good tennis player in US open </s>
<s> tennis player US open </s>
<s> tennis player qualify play US open </s>

w_{n-1}	very	good	tennis	player	in	US	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
US	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

<s>	3
very	1
good	1
tennis	3
player	3
in	1
US	3
open	3
qualify	1
play	1
</s>	3

distinct word = 10

with laplace Smoothing

w_{n-1}	very	good	tennis	player	in	US	open	qualify	play	</s>
<s>	2/13	1/13	3/13							
very										
good										
tennis										
player										
in										
US										
open										
qualify										
play										

Q2) $\frac{P(\text{tennis player US})}{P(\text{tennis player})} = \frac{1}{3}$

$\frac{P(\text{good tennis player})}{P(\text{good tennis})} = \frac{1}{1}$

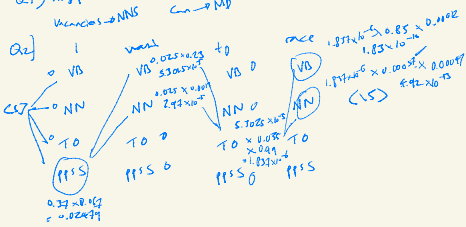
Q3) P("I eat chinese food")
= P(eat | I) * P(chinese | eat) * P(food | Chinese)
= 0.0086 * 0.021 * 0.52

Q4) drawing
- reset n gram probs to 0
- for each sentence in

TUT4

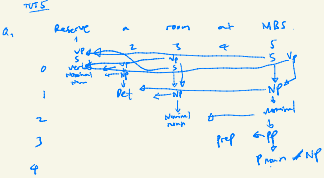
Q1) Singapore → NNP to → CD
Vaccines → NNS Can → MD

Q2)

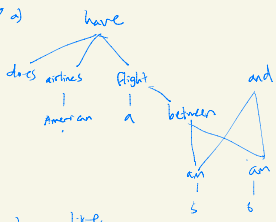


TUT5

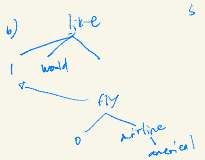
Q1.



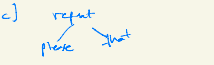
Q2 a)



b)

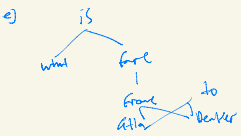


c)



d)

e)



Tutorial 5: Grammar and Parsing

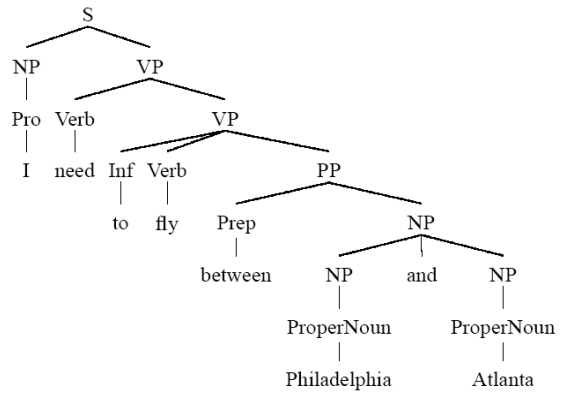
Q1. Consider the L1 grammar used in our lectures.

S→ NP VP	Nominal → Noun	NP→ Verb PP
S→ Aux NP VP	Nominal → Nominal Noun	VP → VP PP
S→ VP	Nominal → Nominal PP	PP→ Preposition NP
NP→ Pronoun	VP→ Verb	
NP→ Proper-Noun	VP→ Verb NP	
NP→ Det Nominal	VP → Verb NP PP	

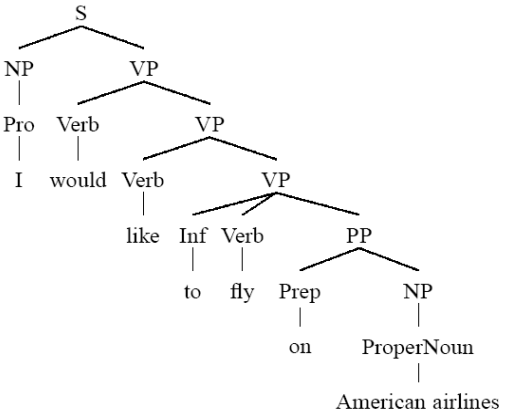
Draw the parse table for the following sentence using the L1 grammar:

Reserve a room at MBS

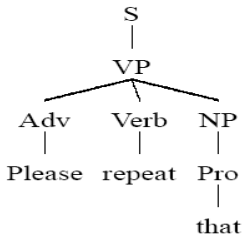
Q2. You are provided with the phrase structures for the following sentences:



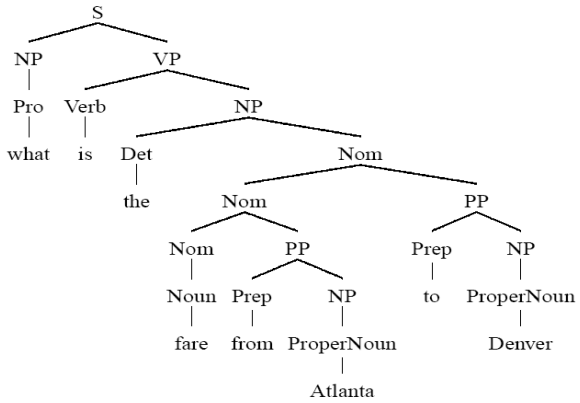
I need to fly between Philadelphia and Atlanta.



I would like to fly on American airlines.



Please repeat that.



What is the fare from Atlanta to Denver?

Q_1 S

Reserve a room at MBS

Revise the L1 grammar Q1 such that the revised grammar can be used to parse the above four sentences.

Q3. Use the following sentences as examples to observe the dependency structures: https://demos.explosion.ai/displacy

a) Do American airlines have a flight between five a.m. and six a.m.?

b) I would like to fly on American airlines.

c) Please repeat that.

d) I need to fly between Philadelphia and Atlanta.

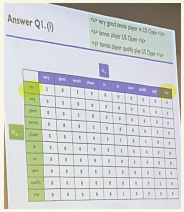
e) What is the fare from Atlanta to Denver?

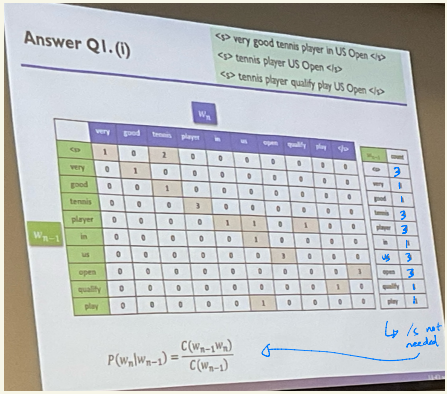
Q4. Draw two dependency structures for the following sentence:

They hid the letter on the wall.

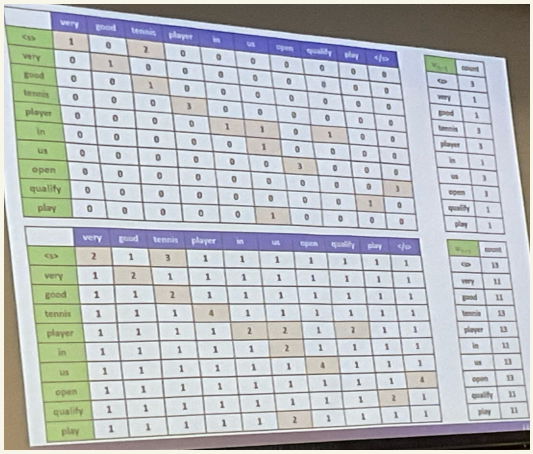
Q5. **Open question**: Try and explore the Rule-based Matcher https://demos.explosion.ai/matcher

Define a pattern and explain why it could be useful for a real-life use case for some text data (e.g., news articles, financial reports, medical reports, product reviews)

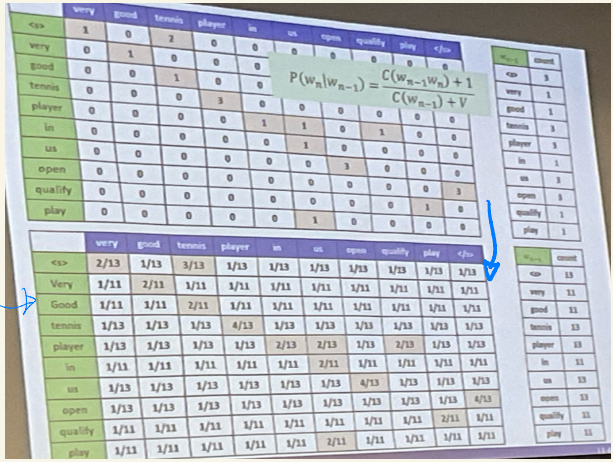




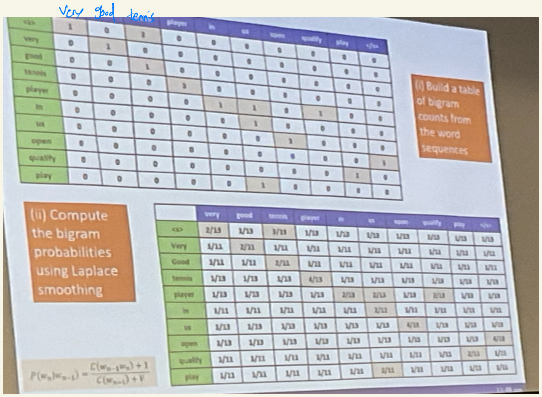
With Laplace Smoothing

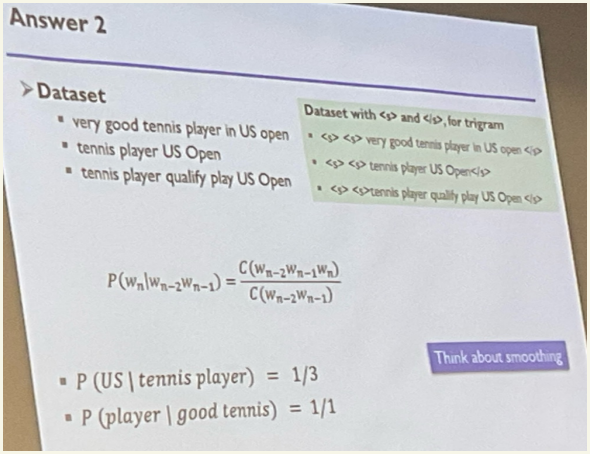


! in exam don't need put decimal



Very good tennis





Answer 3

If not considering <s> and </s>:

- P(I eat Chinese food)
  = P(eat|I) * P(Chinese|I eat) * P(food|I eat Chinese)
- Chain rules: Independence Assumption – bigram
- P(I eat Chinese food)
  = P(eat|I) * P(Chinese|eat) * P(food|Chinese)
  = 0.0036 * 0.021 * 0.52

bigram

Answer 3

In practice, we should consider <s> and </s>:

- P(I eat Chinese food)
  = P(I|<s>) * P(eat|I) * P(Chinese|I eat) * P(food|I eat Chinese)
  * P(</s>|I eat Chinese food)
- P(<s> I eat Chinese food </s>)
  = P(I|<s>) * P(eat|I) * P(Chinese|eat) * P(food|Chinese) * P(</s>|food)
  = ??? * 0.0036 * 0.021 * 0.52 * ???

??? → unknown probabilities from the question.

Wordclap

1) N-gram model cannot be applied to text data with code switching like tweets. False

Answer 4

- Our maximum likelihood estimation is based on training data
- Text data are 'sparse' for the estimation
  - for n-grams that occur a sufficient number of times, it is fine
  - some perfectly acceptable English sequences will be missing from the training corpus
    - 0 probability problem
    - estimate is poor when the counts are small
- e.g., Laplace smoothing and other more advanced smoothing

Question 4

- Why do we need to do smoothing for language model?

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

Answer 5 (The Big Picture)

- Training phase.
  - Reset all n-gram counts to 0.
  - For each sentence in the training data:
    - Update n-gram counts (A).
- Evaluation phase.
  - For each sentence to be evaluated:
    - For each n-gram in the sentence:
      - Call smoothing routine to evaluate probability of n-gram given training counts (B).
  - Compute overall perplexity of evaluation data from n-gram probabilities.

Question 6: for discussion only:

- You are given a text collection of 100GB, and asked to train a bigram language model. You have a computer with 16GB ram and 1TB storage. Think about the best choices (steps) for implementation.
- https://stackoverflow.com/questions/45264957/storing-ngram-model-python
- https://aclanthology.org/W07-0712.pdf
- https://www.vldb.org/pvldb/vol12/p2206-long.pdf
- Resource: https://books.google.com/ngrams/info

Tutorial 4: POS tagging and HMM

Q1. Find tagging errors in each of the following sentences that are tagged with the Penn Treebank tagset. You may get help from online demos of POS tagging services.

1. How/WRB do/MD I/PRP get/VB to/TO Singapore/NN
2. Do/VBP you/PRP have/VB any/DT vacancies/NN
3. This/DT room/NN is/VBZ too/JJ noisy/JJ
4. Can/VB you/PRP give/VB me/PRP another/DT room/NN

Q2. Compute the best tag sequence for "I want to race" using the Viterbi algorithm with the provided HMM parameters, i.e., the transition probability and the word likelihood probabilities.

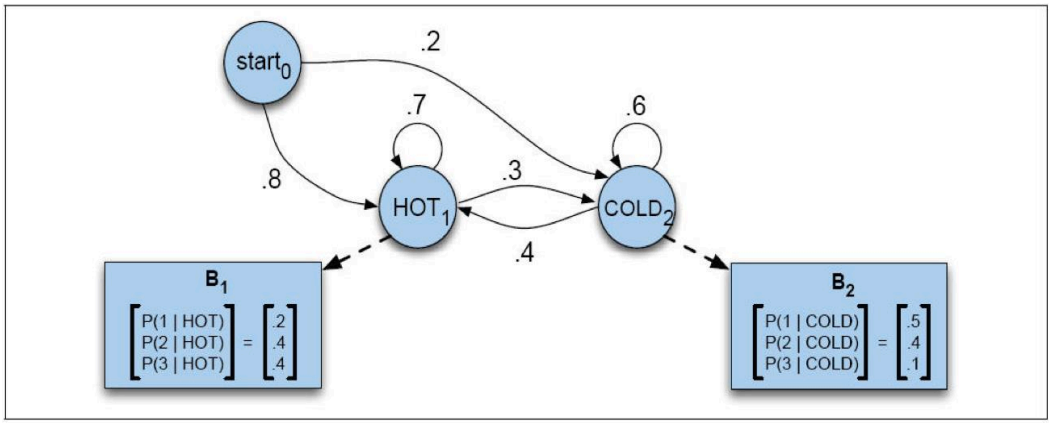
Important

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Q3. Run the Viterbi algorithm with the HMM in the figure below to compute the most likely weather sequences for each of the two following observation sequences. Note: You may consider using HMM packages for computation, e.g., https://pypi.org/project/hmmlearn/

- Sequence: 312312312
- Sequence: 311233112



Answer I

- How/WRB do/VBP I/PRP get/VB to/TO Singapore/NN
  - **Singapore/NNP**
- Do/VBP you/PRP have/VB any/DT vacancies/NN
  - **vacancies/NNS**
- This/DT room/NN is/VBZ too/JJ noisy/JJ
  - **too/RB**
- Can/VB you/PRP give/VB me/PRP another/DT room/NN
  - **Can/MD**

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	"to"	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	ex istential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past partic.	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one's*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlative adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing. or mass noun	*llama*	SYM	symbol	*+, %, &*	WRB	wh-adverb	*how, where*

A sentence has 10 words. You use the HMM and Viterbi algorithm to compute the best tag sequence. After finishing the computation for the 5th word, you will know the POS tags for the first 5 words.

False, need to compute Vertice

Q2

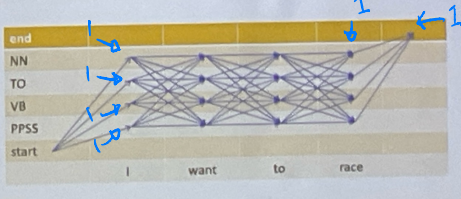
Main Idea

Review

- We also have a matrix.
  - Each column– a time 't' (observation)
  - Each row – a state 'i'
  - For each cell $v_t[i]$, we compute the probability of the **best path** to the cell
- the **Viterbi path probability** at time t for state i
  - there are $|Q|$ number of paths from $t-1$ to $v_t[i]$
  - if we know **the best path** to each cell in $t-1$, or $v_{t-1}[j]$
  - $\arg\max_j v_{t-1}[j] \times P(i|j) \times P(s_t|i)$

Required computations

most imp part



(This figure does not show the backtrace pointers)

1+1+1+1+1+1+1 = 17

Answer 2

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN					$p(NN\|<s>) * p(I\|NN) = 0$
TO					0
VB					0
PPSS					$p(PPSS\|<s>) * p(I\|PPSS)$ $= 0.067 * 0.37 = 0.02479$
start					
	I	want	to	race	

Answer 2

					end
NN	0		1.6×10^{-4}	0	
TO	0	0		$\max(1.6 \times 10^{-4} \times p(TO\|NN), 5.3 \times 10^{-5} \times p(TO\|VB)) * p(to\|TO) =$ $\max(1.6 \times 10^{-4} \times .016, 5.3 \times 10^{-5} \times .035) * .99 = 1.84 \times 10^{-6}$	
VB	0	5.3×10^{-5}	0		
PPSS	0.02479	0	0		
start					
	I	want	to	race	

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN	0	$.02479 \times p(NN\|PPSS) * p(want\|NN) =$ $.02479 \times .0012 \times .000054 =$ 0.00000000160639			
TO	0	0			
VB	0	$.02479 \times p(VB\|PPSS) \times p(want\|VB) =$ $.02479 \times .23 \times .0093 =$ 0.00005302581			
PPSS	0.02479	0			
start					
	I	want	to	race	

Q4. The task of **negation scope detection** is to extract the parts of a sentence that is being negated. For example, in the sentence "I have not submitted my assignment", the negation scope is "submitted my assignment".

Formulate this problem as a sequence labelling task, and discuss how to apply Hidden Markov Model (HMM) to solve this problem. Clearly state the probabilities that need to be learned by the HMM.

Answer 4

- This question is similar to NER, a typical sequence labeling task
- We can use **BIO** label scheme
  - B marks the being of the negation scope.
  - I marks tokens within the negation scope.
  - O marks tokens NOT part of the negation scope.

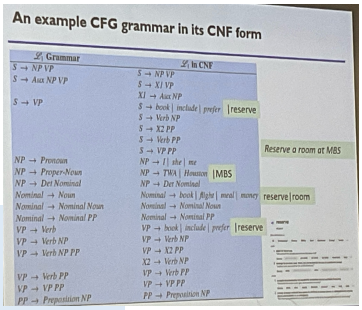
O	O	O	B	I	I	O
I	have	not	submitted	my	assignment	yet

- In the HMM,
  - the hidden states are BIO (example above), and the word sequence is the observed.
  - We learn three probabilities:
    - transition probability between states BIO,
    - observation likelihood of observing a word given a label B, I, or O.
    - The initial probability of BIO in a sentence.

Question 4 → a bit of extension

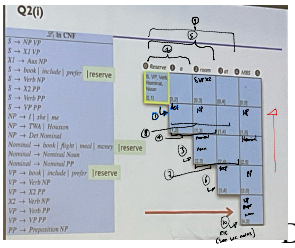
- The task of **negation scope detection** is to extract the parts of a sentence that is being negated. For example, in the sentence "I have not submitted my assignment", the negation scope is "submitted my assignment".
- Consider these sentences:
  - He **seldom** makes mistake.
  - I do **not** know why he is **not** happy.
  - He must be very nervous, but he **denied**. yes, negate
  - He **may or may not** join us for lunch. not a negation

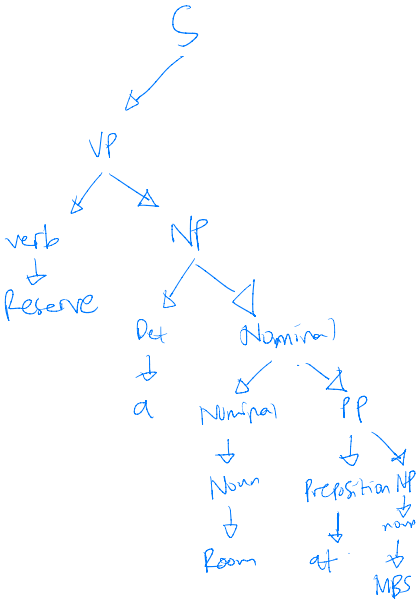
Tutorial 5: Grammar and Parsing



Q1. Consider the L1 grammar used in our lectures.

S→ NP VP	Nominal → Noun	NP→ Verb PP
S→ Aux NP VP	Nominal → Nominal Noun	VP → VP PP
S→ VP	Nominal → Nominal PP	PP→ Preposition NP
NP→ Pronoun	VP→ Verb	
NP→ Proper-Noun	VP→ Verb NP	
NP→ Det Nominal	VP → Verb NP PP	

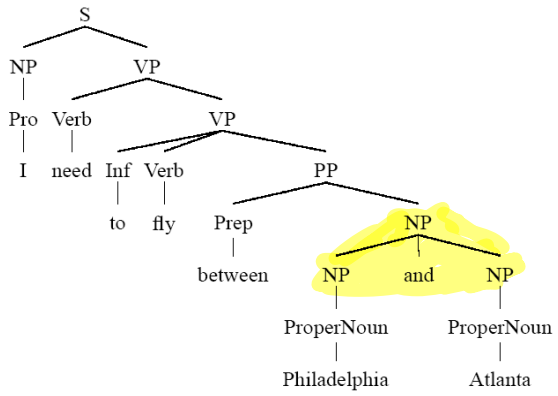




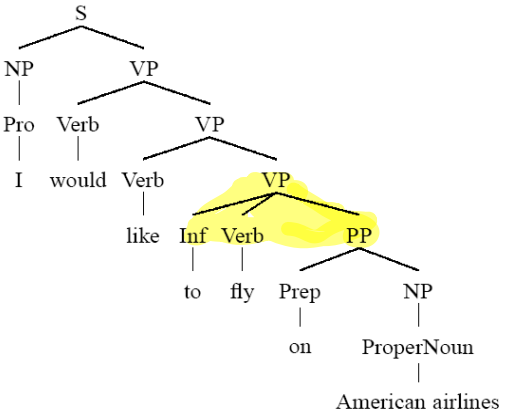
Draw the parse table for the following sentence using the L1 grammar:

Reserve a room at MBS

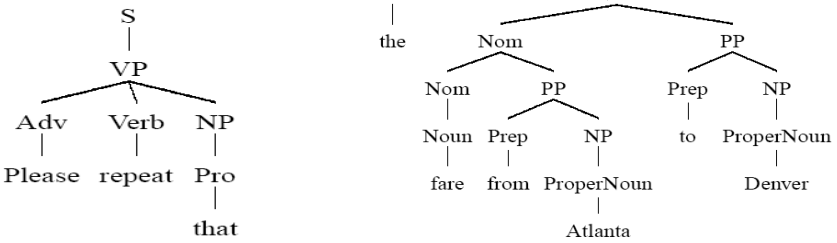
Q2. You are provided with the phrase structures for the following sentences:



I need to fly between Philadelphia and Atlanta.

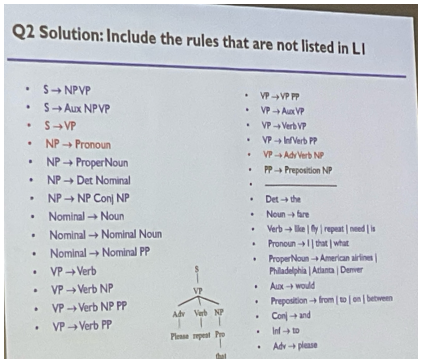


I would like to fly on American airlines.

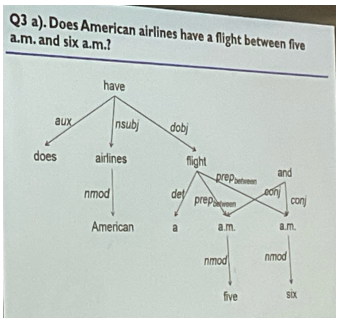


Please repeat that.

What is the fare from Atlanta to Denver?



Revise the L1 grammar Q1 such that the revised grammar can be used to parse the above four sentences.



Q3. Use the following sentences as examples to observe the dependency structures: https://demos.explosion.ai/displacy

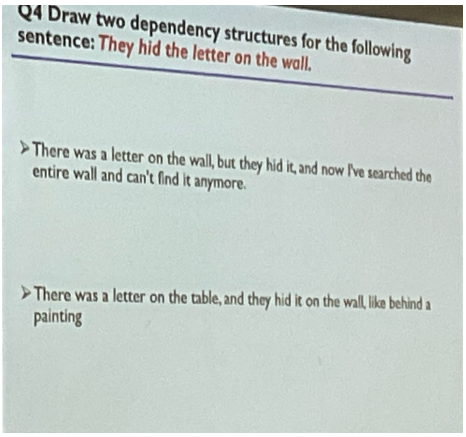
a) Do American airlines have a flight between five a.m. and six a.m.?

b) I would like to fly on American airlines.

c) Please repeat that.

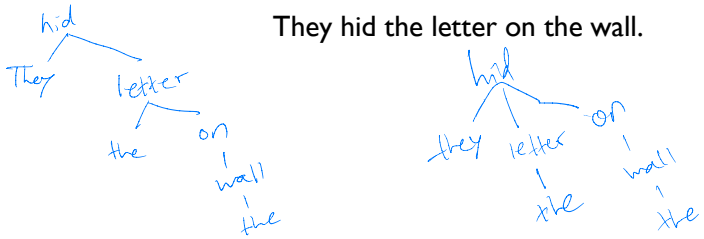
d) I need to fly between Philadelphia and Atlanta.

e) What is the fare from Atlanta to Denver?



Q4. Draw two dependency structures for the following sentence:

They hid the letter on the wall.



Q5. **Open question**: Try and explore the Rule-based Matcher https://demos.explosion.ai/matcher

Define a pattern and explain why it could be useful for a real-life use case for some text data (e.g., news articles, financial reports, medical reports, product reviews)

Online shopping categorize

If an English sentence has 3 valid phrase then it is very likely that the sentence also has 3 different dependency structures

Natural Language Processing

Tutorial 1: Regular Expressions and Text Normalization

Dr. Sun Aixin

Question Q1

- Write regular expressions for the following languages. By "word", we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth. (**HINT**: please consult the book Chapter 2.1 or some websites on regular expressions.)

1. The set of all alphabetic strings;
2. The set of all lower case alphabetic strings ending with a letter *b*;
3. The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug");
4. All strings that start at the beginning of the line with an integer and that end at the end of the line with a word;
5. All strings that have both the word *grotto* and the word *raven* in them (but not, e.g., words like *grottos* that merely *contain* the word *grotto*);

Question 1.1, 1.2

- The set of all alphabetic strings;
  - **[a-zA-Z]+**

- The set of all lower case alphabetic strings ending with a letter b;
  - **[a-z]*b**

Question 1.3

- The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug");
  - **(\b[a-zA-Z]+\b)\s+\1**

- Explanation
  - [a-zA-Z]+ → all alphabetic strings
  - \s → whitespace (space, tab..)
  - \1 → used to refer to back to the first pattern in the expression which is put **inside a parentheses** ( )
    - We may have \2 or \3 to refer to the second and third patterns put inside parentheses.

Question 1.4

- All strings that start at the beginning of the line with an integer and that end at the end of the line with a word
  - **^\d+\b.*\b[a-zA-Z]+$**

- Explanation
  - \d → a digit
  - \b → a word boundary
  - ^, $ → the **beginning** and **end** of a line
  - . → a wildcard expression that matches any single character (except a carriage return)
  - * → Kleene star, zero or more occurrences of the immediate previous character or regular expression
  - .* → any string of characters

Question 1.5

- All strings that have both the word `grotto` and the word `raven` in them (but not, e.g., words like `grottos` that merely contain the word `grotto`)
  - **(.*\bgrotto\b.*\braven\b.*)|(.*\braven\b.*\bgrotto\b.*)**

- Explanation
  - The two words grotto and raven may appear in any order.
  - There could be other strings around the two words

- http://regexr.com/

Question 2

- **Try all your answers** on http://regexr.com/
- You may need to change the textbox to test two cases: the textbox contains one or more matched strings, and the textbox does not contain any matched string.
- What are the errors (e.g., false positive and false negative) have you observed?

Question 2

- The set of all alphabetic strings;
  - **[a-zA-Z]+**
- The set of all lower case alphabetic strings ending with a letter b;
  - **[a-z]*b**
- The set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug");
  - **(\b[a-zA-Z]+\b)\s+\1**

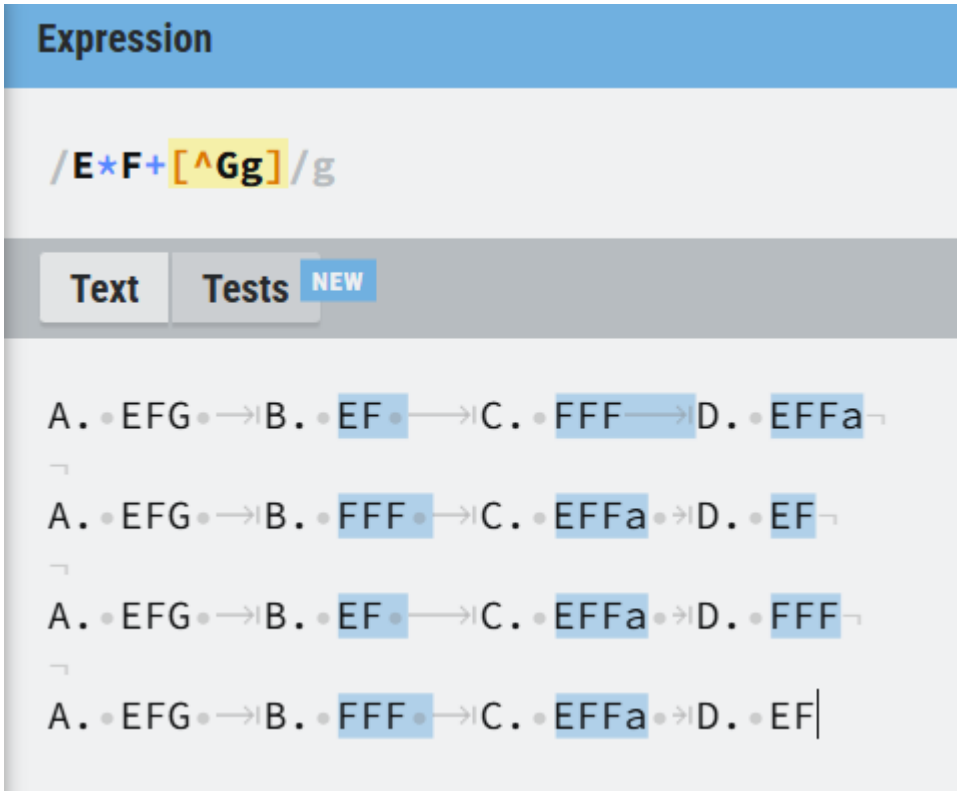
Question 2

- All strings that start at the beginning of the line with an integer and that end at the end of the line with a word
  - **^\d+\b.*\b[a-zA-Z]+$**

- All strings that have both the word `grotto` and the word `raven` in them (but not, e.g., words like `grottos` that merely contain the word `grotto`)
  - **(.*\bgrotto\b.*\braven\b.*)|(.*\braven\b.*\bgrotto\b.*)**

Question 3

- Select all strings that can be matched by regular expression /E*F+[^Gg]/
  - EFG
  - EF
  - FFF
  - EFFa



Question 4

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 1 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.
  - **Note: <u>The costs of Ins, Del, and Sub are predefined based on the application need.</u>**

Target

Source

	#	d	e	a	l
#	**0**	1	2	3	4
i	1				
d	2				
e	3				
a	4				

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1			
d	2	1			
e	3	2			
a	4	3			

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2		
d	2	1	2		
e	3	2	1		
a	4	3	2		

Question 4

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 1 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.

Target

Source

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2		
d	2	1	2		
e	3	2	1		
a	4	3	2		

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2	3	
d	2	1	2	3	
e	3	2	1	2	
a	4	3	2	1	

	#	d	e	a	l
#	**0**	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2

Question 4

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1]+\begin{cases} 1 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 1) of "idea" to "deal". Show your work.

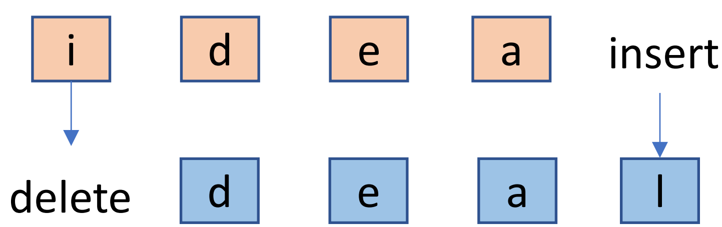
Target

Source

	#	d	e	a	l
#	0	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2

	#	d	e	a	l
#	0	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2

	#	d	e	a	l
#	0	1	2	3	4
i	1	1	2	3	4
d	2	1	2	3	4
e	3	2	1	2	3
a	4	3	2	1	2



Question 5

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 2 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 2) of two sentences "computed the edit distance" to "the edit distance is computed". Show your work and show the alignment between the two strings.
  - If similar questions appear in exam, the requirement will be made clear, whether the edit distance is to be computed at character level or at word level.
  - We use the first character to represent each word, for simplicity and clarity.

Target

Source

	#	T	E	D	I	C
#	**0**	1	2	3	4	5
C	1					
T	2					
E	3					
D	4					

	#	T	E	D	I	C
#	**0**	1	2	3	4	5
C	1	2	3	4	5	4
T	2	1	2	3	4	5
E	3	2	1	2	3	4
D	4	3	2	1	2	3

Question 5

$$D[i,j] = \min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1] + \begin{cases} 2 & \text{if } source[i] \neq target[j] \\ 0; & \text{if } source[i] = target[j] \end{cases} \end{cases}$$

- Compute the edit distance (using insertion cost 1, deletion cost 1, substitution cost 2) of two sentences "computed the edit distance" to "the edit distance is computed". Show your work and show the alignment between the two strings.

computed **the edit distance**

the edit distance is computed

Target

Source

	#	T	E	D	I	C
#	**0**	1	2	3	4	5
C	1	2	3	4	5	4
T	2	1	2	3	4	5
E	3	2	1	2	3	4
D	4	3	2	1	2	3

	#	T	E	D	I	C
#	**0**	1	2	3	4	5
C	1	2	3	4	5	4
T	2	1	2	3	4	5
E	3	2	1	2	3	4
D	4	3	2	1	2	3

Natural Language Processing

Tutorial 2: Text Normalization

Dr. Sun Aixin

Question 1

- Consider the following word segmentation algorithm in the lecture notes:

- Given a lexicon of Chinese, and a string
  - Start a pointer at the beginning of the string
  - Find the longest word in dictionary that matches the string starting at pointer
  - Move the pointer over the word in string
  - Goto2

- Following the algorithm, you perhaps end up with failing to segment a string, if you cannot find a matching.

Question 1 (cont)

- Example
  - String to segment: thetablesdownthere
  - lexicon: the table down there bled own.
- Discuss how to fix the above problem.

Answer Q1

- Start a pointer at the beginning of the string
- Find the longest word in dictionary that matches the string starting at pointer
  - If matched, move the pointer over the word in string
  - If no word is matched, skip to the next letter
- Goto2

  - String to segment: thetablesdownthere
  - lexicon: the table down there bled own.

Answer 1 (a bit more processing)

- Start a pointer at the beginning of the string
- Find the longest word in dictionary that matches the string starting at pointer
  - If matched,
    - Run morphological analysis from the beginning of the word
    - Move the pointer over the morphologically matched part of the string
  - If no word is matched, skip to the next letter
- Goto2

  - String to segment: thetablesdownthere
  - lexicon: the table down there bled own.

Question 2

- Try the tokenization demo on
  - http://text-processing.com/demo/tokenize/
  - (you may use other tokenizer APIs).
- Discuss your findings based on the output of different tokenizers.

- https://textanalysisonline.com/

Question 3

- Try the stemmer demo on
  - http://text-processing.com/demo/stem/
  - (or you may use other stemmer APIs).
- Discuss your findings based on the output of the stemmers.
- https://textanalysisonline.com/

Question 4

- Write a program to do the following tasks:
  - **Download** the Web page of a given link and **extract** the text content of the page
  - **Split** the text into sentences and **count** sentences
  - **Split** the text into tokens and **count** token types
  - **Find** lemmas (or stems) of the tokens and **count** lemma types
  - Do **stemming** on the tokens and **count** unique stemmed tokens

Answer 4

- Example URL:
  - https://en.wikipedia.org/wiki/Natural_language_processing
- urllib
  - https://docs.python.org/3/library/urllib.html
- NLTK
  - https://www.nltk.org/
- Beautiful Soup
  - https://www.crummy.com/software/BeautifulSoup/bs4/doc/#porting-code-to-bs4

Alternative libraries:
StanfordNLP
OpenNLP
spaCy
AllenNLP

Sample code

```
import urllib.request
import nltk
from bs4 import BeautifulSoup

with urllib.request.urlopen ('https://en.wikipedia.org/wiki/Natural_language_processing') as response:
    html=response.read()

text = BeautifulSoup(html, "lxml").get_text()
sentences = nltk.tokenize.sent_tokenize(text)
print ('Number of sentences: '+ str(len(sentences)))

tokens= nltk.tokenize.word_tokenize(text)

print ('Number of tokens: '+ str(len(tokens)))

token_types = list(set(tokens))

print ('Number of token types: '+ str(len(token_types)))

wnl=nltk.stem.WordNetLemmatizer()

stemmer = nltk.stem.porter.PorterStemmer()
lemma_types= set()
stemmed_types= set()

for token_type in token_types:
    lemma_types.add(wnl.lemmatize(token_type))
    stemmed_types.add(stemmer.stem(token_type))

print ('Number of lemma types: '+ str(len(lemma_types)))
print ('Number of stemmed types: '+ str(len(stemmed_types)))
```

Download the Web page of a given link and **extract** the text content of the page

Split the text into sentences and **count** sentences

Split the text into tokens and **count** token types

Find lemmas (or stems) of the tokens and **count** lemma types

Do **stemming** on the tokens and **count** unique stemmed tokens

Sample code

- text = BeautifulSoup(html, "lxml").get_text()

Beautiful Soup Documentation

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

This document covers Beautiful Soup version 4.8.1. The examples in this documentation should work the same way in Python 2.7 and Python 3.2.

https://beautiful-soup-4.readthedocs.io/en/latest/

Sample code

https://www.nltk.org/api/nltk.tokenize.html

- sentences = nltk.tokenize.sent_tokenize(text)
- tokens= nltk.tokenize.word_tokenize(text)

- nltk.tokenize
  - nltk.tokenize.api
  - nltk.tokenize.casual
  - nltk.tokenize.destructive
  - nltk.tokenize.legality_principle
  - nltk.tokenize.mwe
  - nltk.tokenize.nist
  - nltk.tokenize.punkt
  - nltk.tokenize.regexp
  - nltk.tokenize.repp
  - nltk.tokenize.sexpr
  - nltk.tokenize.simple
  - nltk.tokenize.sonority_sequencing
  - nltk.tokenize.stanford
  - nltk.tokenize.stanford_segmenter
  - nltk.tokenize.texttiling
  - nltk.tokenize.toktok
  - nltk.tokenize.treebank
  - nltk.tokenize.util

nltk.tokenize package

NLTK Tokenizer Package

Tokenizers divide strings into lists of substrings. For example, tokenizers can be used to find the words and punctuation in a string:

```
>>> from nltk.tokenize import word_tokenize
>>> s = '''Good muffins cost $3.88\nin New York.  Please buy me
... two of them.\n\nThanks.'''
>>> word_tokenize(s)
['Good', 'muffins', 'cost', '$', '3.88', 'in', 'New', 'York', '.',
'Please', 'buy', 'me', 'two', 'of', 'them', '.', 'Thanks', '.']
```

This particular tokenizer requires the Punkt sentence tokenization models to be installed. NLTK also provides a simpler, regular-expression based tokenizer, which splits text on whitespace and punctuation:

```
>>> from nltk.tokenize import wordpunct_tokenize
>>> wordpunct_tokenize(s)
['Good', 'muffins', 'cost', '$', '3', '.', '88', 'in', 'New', 'York', '.',
'Please', 'buy', 'me', 'two', 'of', 'them', '.', 'Thanks', '.']
```

We can also operate at the level of sentences, using the sentence tokenizer directly as follows:

```
>>> from nltk.tokenize import sent_tokenize, word_tokenize
>>> sent_tokenize(s)
['Good muffins cost $3.88\nin New York.', 'Please buy me\ntwo of them.', 'Thanks.']
>>> [word_tokenize(t) for t in sent_tokenize(s)]
[['Good', 'muffins', 'cost', '$', '3.88', 'in', 'New', 'York', '.'],
['Please', 'buy', 'me', 'two', 'of', 'them', '.'], ['Thanks', '.']]
```

Sample code

https://www.nltk.org/api/nltk.stem.html
https://www.nltk.org/api/nltk.stem.wordnet.html
https://web.stanford.edu/~jurafsky/slp3/

- wnl=nltk.stem.WordNetLemmatizer()
- stemmer = nltk.stem.porter.PorterStemmer()

nltk.stem package

NLTK Stemmers

Interfaces used to remove morphological affixes from words, leaving only the word stem. Stemming algorithms aim to remove those affixes required for eg. grammatical role, tense, derivational morphology leaving only the stem of the word. This is a difficult problem due to irregular words (eg. common verbs in English), complicated morphological rules, and part-of-speech and sense ambiguities (eg. `ceil-` is not the stem of `ceiling`).

StemmerI defines a standard interface for stemmers.

Submodules

- nltk.stem.api module
- nltk.stem.arlstem module
- nltk.stem.arlstem2 module
- nltk.stem.cistem module
- nltk.stem.isri module
- nltk.stem.lancaster module
- nltk.stem.porter module
- nltk.stem.regexp module
- nltk.stem.rslp module
- nltk.stem.snowball module
- nltk.stem.util module
- nltk.stem.wordnet module

nltk.stem.wordnet module

class nltk.stem.wordnet.WordNetLemmatizer [source]

Bases: `object`

WordNet Lemmatizer

Lemmatize using WordNet's built-in morphy function. Returns the input word unchanged if it cannot be found in WordNet.

```
>>> from nltk.stem import WordNetLemmatizer
>>> wnl = WordNetLemmatizer()
>>> print(wnl.lemmatize('dogs'))
dog
>>> print(wnl.lemmatize('churches'))
church
>>> print(wnl.lemmatize('aardwolves'))
aardwolf
>>> print(wnl.lemmatize('abaci'))
abacus
>>> print(wnl.lemmatize('hardrock'))
hardrock
```

Q5: Open-ended Question, for discussion only.

- In social media (e.g., forums), online users often use informal names or references when mentioning products. Below are example sentences discussing mobile phones, where the words highlighted in bold are the phones being referred to, and the [bracketed text] indicates their official product names.

1.	True, **Desire** [HTC Desire] might be better if compared to **X10** [Sony Ericsson Xperia X10] but since I am using **HD2** [HTC HD2], it will be a little boring to use back HTC ...
2.	I just wanna know what problems do users face on the **OneX** [HTC One X]... of course I know that knowing the problems on **one x** [HTC One X] doesn't mean knowing the problems on **s3** [Samsung Galaxy SIII]
3.	Still prefer **ip 5** [Apple iPhone 5] then **note 2** [Samsung Galaxy Note II]...
4.	oh, the mono rich recording at **920** [Nokia Lumia 920] no better than stereo rich recording at **808** [Nokia 808 PureView].

Q5: Open-ended Question, for discussion only.

- The table below shows the number of users who have mentioned a phone using a specific name in a forum.

Name variation	#users	Name variation	#users
1. galaxy s3	553	14. lte s3	46
2. s3 lte	343	15. galaxy s3 lte	45
3. samsung galaxy s3	284	16. s3 non lte	32
4. s iii	242	17. samsung galaxy siii	32
5. galaxy s iii	225	18. sgs 3	27
6. samsung s3	219	19. samsung galaxy s3 lte	22
7. sgs3	187	20. sg3	21
8. siii	149	21. gsiii	16
9. samsung galaxy s iii	145	22. samsung galaxy s3 i9300	15
10. i9300	120	23. samsung i9300 galaxy s iii	13
11. gs3	82	24. s3 4g	11
12. galaxy siii	61	25. 3g s3	11
13. i9305	52	–	

Q5: Open-ended Question, for discussion only.

- Task: Assume that we have successfully identified the phone mentions (e.g., 's3 lte,' 'sgs3'). How can we **normalize these mentions to their formal names**?

1. True, **Desire** [HTC Desire] might be better if compared to **X10** [Sony Ericsson Xperia X10] but since I am using **HD2** [HTC HD2], it will be a little boring to use back HTC ...
2. I just wanna know what problems do users face on the **OneX** [HTC One X]... of course I know that knowing the problems on **one x** [HTC One X] doesn't mean knowing the problems on **s3** [Samsung Galaxy SIII]
3. Still prefer **ip 5** [Apple iPhone 5] then **note 2** [Samsung Galaxy Note II]...
4. oh, the mono rich recording at **920** [Nokia Lumia 920] no better than stereo rich recording at **808** [Nokia 808 PureView].

Name variation	#users	Name variation	#users
1. galaxy s3	553	14. lte s3	46
2. s3 lte	343	15. galaxy s3 lte	45
3. samsung galaxy s3	284	16. s3 non lte	32
4. s iii	242	17. samsung galaxy siii	32
5. galaxy s iii	225	18. sgs 3	27
6. samsung s3	219	19. samsung galaxy s3 lte	22
7. sgs3	187	20. sg3	21
8. siii	149	21. gsiii	16
9. samsung galaxy s iii	145	22. samsung galaxy s3 i9300	15
10. i9300	120	23. samsung i9300 galaxy s iii	13
11. gs3	82	24. s3 4g	11
12. galaxy siii	61	25. 3g s3	11
13. i9305	52	–	

Natural Language Processing

Tutorial 3: N-gram and Language Model

Dr. Sun Aixin

Question 1

- Given the following three word sequences (the corpus)
  - very good tennis player in US Open
  - tennis player US Open
  - tennis player qualify play US Open

- (i) Build a table of bigram counts from the word sequences
- (ii) Compute the bigram probabilities using Laplace smoothing

Bigram model

Review

- With bigram model $\boldsymbol{P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})}$
  - Our example

$$P(w|h) = P(it|I\ will\ make) \approx \boldsymbol{P(it|make)}$$

  - $P(w_{1:n}) = \prod_{k=1}^{n} P(w_k|w_{1:k-1}) \approx \prod_{k=1}^{n} P(w_k|w_{k-1})$

- Now, how to compute $\boldsymbol{P(w_n|w_{n-1})}$, like $\boldsymbol{P(it|make)}$?
  - Estimate bigram probabilities by **maximum likelihood estimation** or MLE
  - We estimate $P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$ where $C(\cdot)$ is the count, or frequency

make decisions.
make sure …
make it right
make it happen
make toys.

$C(make) = 5$
$C(make\ it) = 2$

$P(it|make) = 0.4$

Answer Q1. (i)

- Given the corpus, build a table of bigram counts from the word sequences
  - very good tennis player in US Open
  - tennis player US Open
  - tennis player qualify play US Open

- We should consider the **sentence boundaries** as tokens.
  - <s> very good tennis player in US Open </s>
  - <s> tennis player US Open </s>
  - <s> tennis player qualify play US Open </s>

- Both <s> and </s> are counted as tokens.

make decisions
make sure
make it right
make it happen
make toys

$C(make) = 5$
$C(make\ it) = 2$

$P(it|make) = 0.4$

Answer Q1. (i)

<s> very good tennis player in US Open </s>

<s> tennis player US Open </s>

<s> tennis player qualify play US Open </s>

w_n

w_{n-1}	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

Answer Q1. (i)

<s> very good tennis player in US Open </s>
<s> tennis player US Open </s>
<s> tennis player qualify play US Open </s>

w_n

w_{n-1}	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1}	count
<s>	3
very	1
good	1
tennis	3
player	3
in	1
us	3
open	3
qualify	1
play	1

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1}	count
<s>	3
very	1
good	1
tennis	3
player	3
in	1
us	3
open	3
qualify	1
play	1

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2	1	3	1	1	1	1	1	1	1
very	1	2	1	1	1	1	1	1	1	1
good	1	1	2	1	1	1	1	1	1	1
tennis	1	1	1	4	1	1	1	1	1	1
player	1	1	1	1	2	2	1	2	1	1
in	1	1	1	1	1	2	1	1	1	1
us	1	1	1	1	1	1	4	1	1	1
open	1	1	1	1	1	1	1	1	1	4
qualify	1	1	1	1	1	1	1	1	2	1
play	1	1	1	1	1	2	1	1	1	1

w_{n-1}	count
<s>	13
very	11
good	11
tennis	13
player	13
in	11
us	13
open	13
qualify	11
play	11

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0					
good	0	0	1	0	0					
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1}	count
<s>	3
very	1
good	1
tennis	3
player	3
in	1
us	3
open	3
qualify	1
play	1

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2/13	1/13	3/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
Very	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
Good	1/11	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
tennis	1/13	1/13	1/13	4/13	1/13	1/13	1/13	1/13	1/13	1/13
player	1/13	1/13	1/13	1/13	2/13	2/13	1/13	2/13	1/13	1/13
in	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11
us	1/13	1/13	1/13	1/13	1/13	1/13	4/13	1/13	1/13	1/13
open	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	4/13
qualify	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	2/11	1/11
play	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11

w_{n-1}	count
<s>	13
very	11
good	11
tennis	13
player	13
in	11
us	13
open	13
qualify	11
play	11

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1}	count
<s>	3
very	1
good	1
tennis	3
player	3
in	1
us	3
open	3
qualify	1
play	1

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2/13	1/13	3/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
Very	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
Good	1/11	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
tennis	1/13	1/13	1/13	4/13	1/13	1/13	1/13	1/13	1/13	1/13
player	1/13	1/13	1/13	1/13	2/13	2/13	1/13	2/13	1/13	1/13
in	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11
us	1/13	1/13	1/13	1/13	1/13	1/13	4/13	1/13	1/13	1/13
open	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	4/13
qualify	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	2/11	1/11
play	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11

w_{n-1}	count
<s>	13
very	11
good	11
tennis	13
player	13
in	11
us	13
open	13
qualify	11
play	11

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

(i) Build a table of bigram counts from the word sequences

(ii) Compute the bigram probabilities using Laplace smoothing

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2/13	1/13	3/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
Very	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
Good	1/11	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
tennis	1/13	1/13	1/13	4/13	1/13	1/13	1/13	1/13	1/13	1/13
player	1/13	1/13	1/13	1/13	2/13	2/13	1/13	2/13	1/13	1/13
in	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11
us	1/13	1/13	1/13	1/13	1/13	1/13	4/13	1/13	1/13	1/13
open	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	4/13
qualify	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	2/11	1/11
play	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11

Question 2

- Write out the equation for trigram probability estimation, and use the equation to compute the trigram probability for $P\ (US|\ tennis\ player)$ and $P(player\ |\ good\ tennis)$ according to the corpus given in Q1.

- $P(w_n|w_{n-2}w_{n-1}) = \frac{C(w_{n-2}w_{n-1}w_n)}{C(w_{n-2}w_{n-1})}$

Answer 2

- **Dataset**
  - very good tennis player in US open
  - tennis player US Open
  - tennis player qualify play US Open

Dataset with <s> and </s>, for trigram

- <s> <s> very good tennis player in US open </s>
- <s> <s> tennis player US Open</s>
- <s> <s>tennis player qualify play US Open </s>

$$P(w_n|w_{n-2}w_{n-1}) = \frac{C(w_{n-2}w_{n-1}w_n)}{C(w_{n-2}w_{n-1})}$$

- $P\ (US \mid tennis\ player) = 1/3$
- $P\ (player \mid good\ tennis) = 1/1$

Think about smoothing

Question 3

- Given the bigram probability in the following table, compute the probability of "I eat Chinese food" by using the table. Explain how you compute the probability.
- **State your assumptions** and if more probability values are needed, you may use random values.

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Answer 3

If not considering <s> and </s>:

- $P(I\ eat\ Chinese\ food)$
  $= P(eat|I) * P(Chinese|I\ eat) * P(food|I\ eat\ Chinese)$

- Chain rules: Independence Assumption – bigram

- $P(I\ eat\ Chinese\ food)$
  $= P(eat|I) * P(Chinese|eat) * P(food|Chinese)$
  $= 0.0036 * 0.021 * 0.52$

Answer 3

In practice, we should consider <s> and </s>:

- $P(I\ eat\ Chinese\ food)$
  $= P(I|<s>) * P(eat|I) * P(Chinese|I\ eat) * P(food|I\ eat\ Chinese)$
  $* P(</s>|I\ eat\ Chinese\ food)$

- $P(<s>\ I\ eat\ Chinese\ food\ </s>)$
  $= P(I|<s>) * P(eat|I) * P(Chinese|eat) * P(food|Chinese) * P(</s>|food)$
  $= ??? * 0.0036 * 0.021 * 0.52 * ???$

??? → unknown probabilities from the question.

Question 4

- Why do we need to do smoothing for language model?

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$

Answer 4

- Our maximum likelihood estimation is based on training data

- Text data are 'sparse' for the estimation
  - for n-grams that occur a sufficient number of times, it is fine
  - some perfectly acceptable English sequences will be missing from the training corpus
    - 0 probability problem
    - estimate is poor when the counts are small

- e.g., Laplace smoothing and other more advanced smoothing

Question 5

- Given some text, what are the general steps to collect all counts needed for building an n-gram language model?

Answer 5 (The Big Picture)

- Training phase.
  - Reset all n-gram counts to 0.
  - For each sentence in the training data:
    - Update n-gram counts (A).

- Evaluation phase.
  - For each sentence to be evaluated:
    - For each n-gram in the sentence:
      - Call smoothing routine to evaluate probability of n-gram given training counts (B).
  - Compute overall perplexity of evaluation data from n-gram probabilities.

Question 6: for discussion only:

- You are given a text collection of 100GB, and asked to train a bigram language model. You have a computer with 16GB ram and 1TB storage. Think about the best choices (steps) for implementation.

- https://stackoverflow.com/questions/45264957/storing-ngram-model-python
- https://aclanthology.org/W07-0712.pdf
- https://www.vldb.org/pvldb/vol12/p2206-long.pdf

- Resource: https://books.google.com/ngrams/info

Resources

- Lucene http://lucene.apache.org/core/7_4_0/index.html
- OpenNLP https://opennlp.apache.org/
- Stanford NLP https://nlp.stanford.edu/
- spaCy https://spacy.io/
- NLTK https://www.nltk.org/

Natural Language Processing

Tutorial 4: POS tagging and HMM

Dr. Sun Aixin

Question 1

- Find one tagging error in each of the following sentences that are tagged with the Penn Treebank tagset:

1. How/WRB do/VBP I/PRP get/VB to/TO Singapore/NN
2. Do/VBP you/PRP have/VB any/DT vacancies/NN
3. This/DT room/NN is/VBZ too/JJ noisy/JJ
4. Can/VB you/PRP give/VB me/PRP another/DT room/NN

Penn TreeBank POS Tagset

Review

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	"to"	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one's*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	+,%, &	WRB	wh-adverb	*how, where*

Answer 1

- How/WRB do/VBP I/PRP get/VB to/TO Singapore/NN
  - **Singapore/NNP**
- Do/VBP you/PRP have/VB any/DT vacancies/NN
  - **vacancies/NNS**
- This/DT room/NN is/VBZ too/JJ noisy/JJ
  - **too/RB**
- Can/VB you/PRP give/VB me/PRP another/DT room/NN
  - **Can/MD**

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	*and, but, or*	NNP	proper noun, sing.	*IBM*	TO	"to"	*to*
CD	cardinal number	*one, two*	NNPS	proper noun, plu.	*Carolinas*	UH	interjection	*ah, oops*
DT	determiner	*a, the*	NNS	noun, plural	*llamas*	VB	verb base	*eat*
EX	existential 'there'	*there*	PDT	predeterminer	*all, both*	VBD	verb past tense	*ate*
FW	foreign word	*mea culpa*	POS	possessive ending	*'s*	VBG	verb gerund	*eating*
IN	preposition/ subordin-conj	*of, in, by*	PRP	personal pronoun	*I, you, he*	VBN	verb past participle	*eaten*
JJ	adjective	*yellow*	PRP$	possess. pronoun	*your, one's*	VBP	verb non-3sg-pr	*eat*
JJR	comparative adj	*bigger*	RB	adverb	*quickly*	VBZ	verb 3sg pres	*eats*
JJS	superlative adj	*wildest*	RBR	comparative adv	*faster*	WDT	wh-determ.	*which, that*
LS	list item marker	*1, 2, One*	RBS	superlatv. adv	*fastest*	WP	wh-pronoun	*what, who*
MD	modal	*can, should*	RP	particle	*up, off*	WP$	wh-possess.	*whose*
NN	sing or mass noun	*llama*	SYM	symbol	+,%, &	WRB	wh-adverb	*how, where*

Question 2

- Compute the best tag sequence for "I want to race" using the Viterbi algorithm with the provided HMM parameters, i.e., the transition probability and the word likelihood probabilities

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

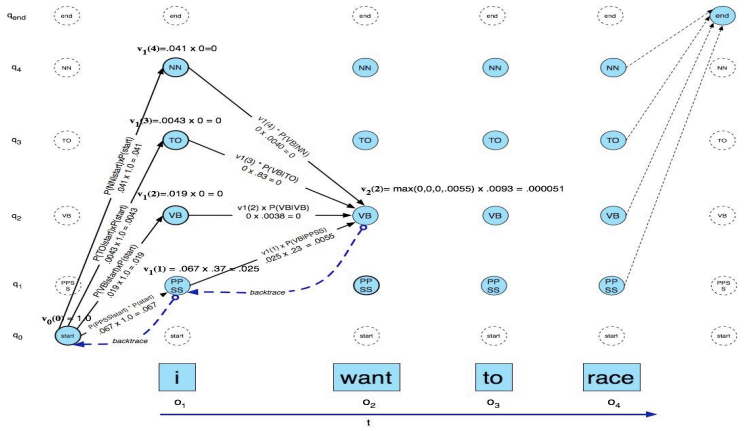
	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

Main Idea

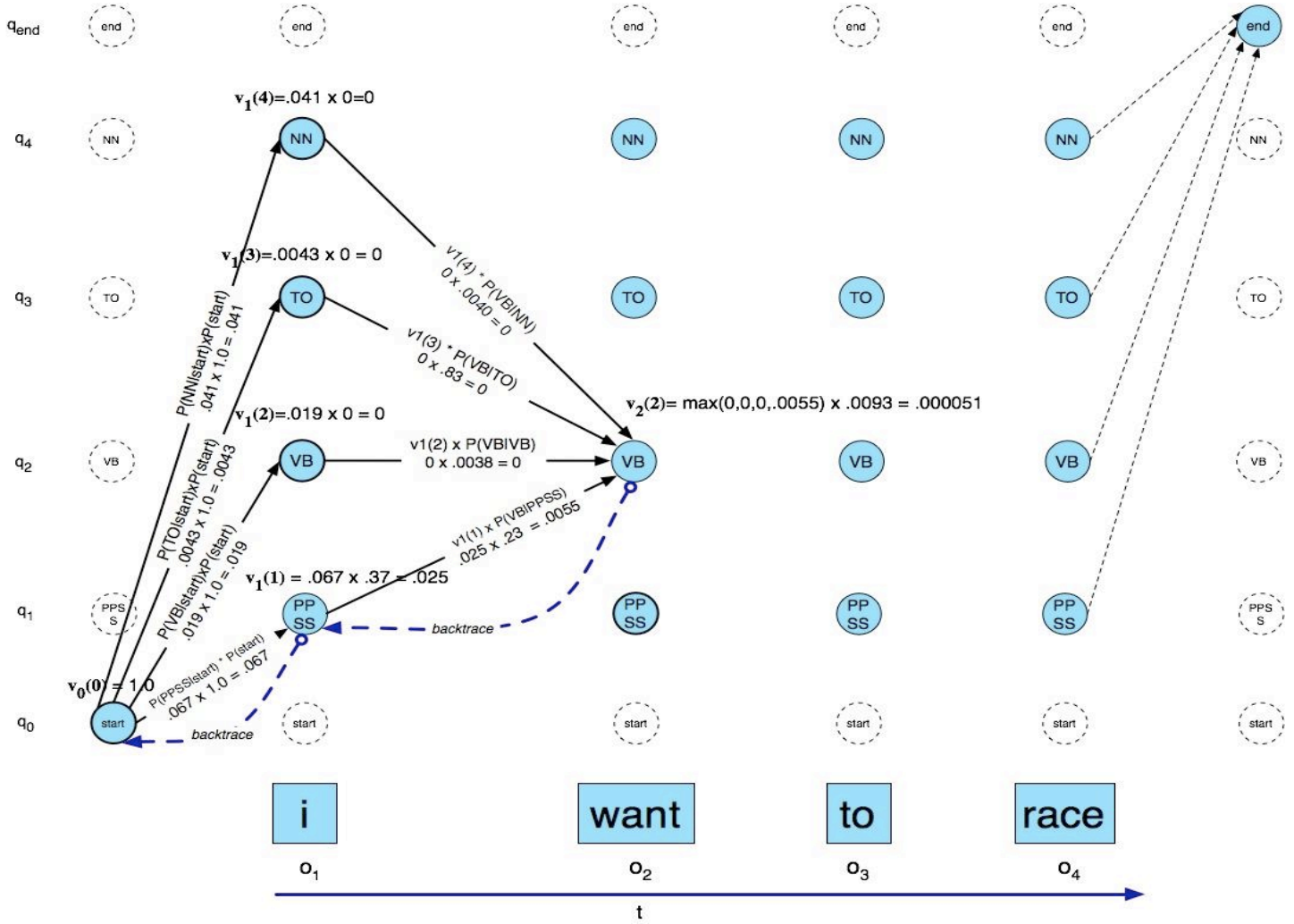
Review

- We also have a matrix.
  - Each column– a time 't' (observation)
  - Each row – a state 'i'
  - For each cell $v_t[i]$, we compute the probability of the **best path** to the cell

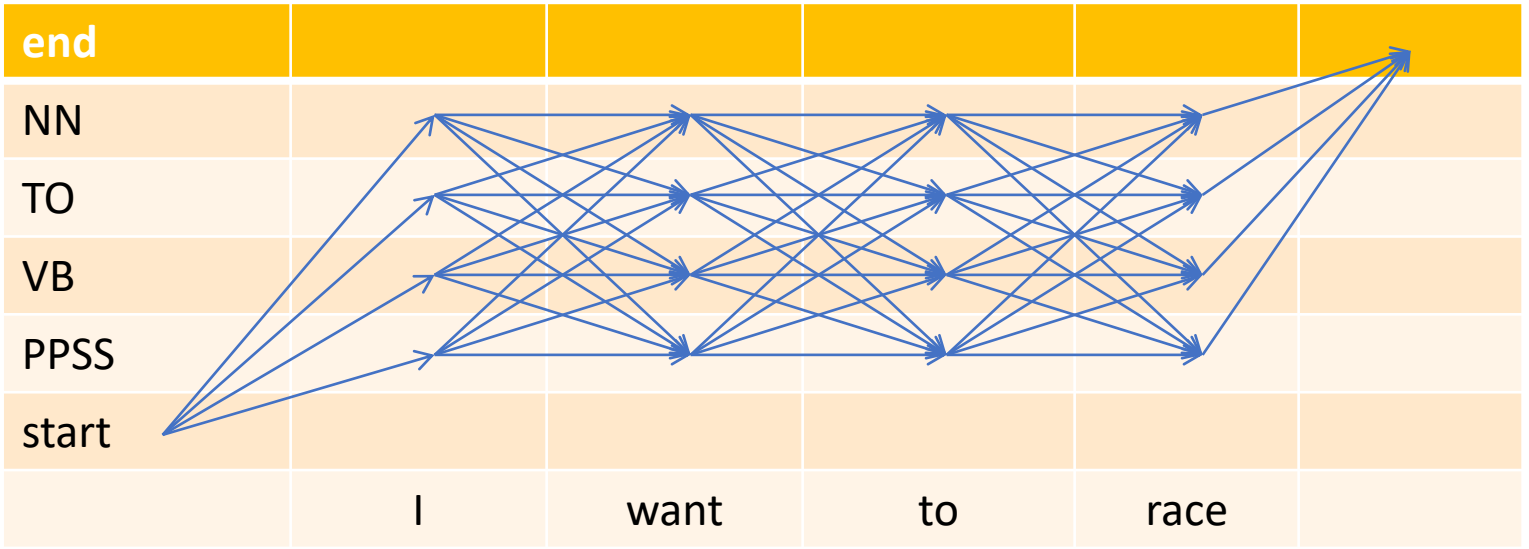
- the **Viterbi path probability** at time t for state i
  - there are $|Q|$ number of paths from $t-1$ to $v_t[i]$
  - if we know **the best path** to each cell in $t-1$, or $v_{t-1}[j]$
  - $\arg\max_j v_{t-1}[j] \times P(i|j) \times P(s_t|i)$



Viterbi Example



Required computations



(This figure does not show the backtrace pointers)

Answer 2

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN	$p(NN\| < s >) * p(I\|NN) = 0$				
TO	0				
VB	0				
PPSS	$p(PPSS\| < s >) * p(I\|PPSS) = 0.067 * 0.37 = 0.02479$				
start					
	I	want	to	race	

Answer 2

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN	0	$.02479 \times p(NN\|PPSS) * p(want\|NN) = .02479 \times .0012 \times .000054 = 0.00000000160639$			
TO	0	0			
VB	0	$.02479 \times p(VB\|PPSS) \times p(want\|VB) = .02479 \times .23 \times .0093 = 0.00005302581$			
PPSS	0.02479	0			
start					
	I	want	to	race	

Answer 2

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN	0	1.6×10^{-9}	0		
TO	0	0	$\max\left(1.6 \times 10^{-9} \times p(TO\|NN), 5.3 \times 10^{-5} \times p(TO\|VB)\right) * p(to\|TO) = \max(1.6 \times 10^{-9} \times .016, 5.3 \times 10^{-5} \times .035) * .99 = 1.84 \times 10^{-6}$		
VB	0	5.3×10^{-5}	0		
PPSS	0.02479	0	0		
start					
	I	want	to	race	

Answer 2

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN	0	1.6×10^{-9}	0	$1.84 \times 10^{-6} \times p(NN\|TO) \times p(race\|NN) = 1.84 \times 10^{-6} \times .00047 \times .00057 = 4.92 \times 10^{-14}$	
TO	0	0	1.84×10^{-6}	0	
VB	0	5.3×10^{-5}	0	$1.84 \times 10^{-6} \times p(VB\|TO) \times p(race\|VB) = 1.84 \times 10^{-6} \times .83 \times .00012 = 1.83 \times 10^{-10}$	
PPSS	0.02479	0	0	0	
start					
	I	want	to	race	

Answer 2

	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN	0	1.6×10^{-9}	0	$1.84 \times 10^{-6} \times p(NN\|TO) \times p(race\|NN) = 1.84 \times 10^{-6} \times .00047 \times .00057 = 4.92 \times 10^{-14}$	
TO	0	0	1.84×10^{-6}	0	
VB	0	5.3×10^{-5}	0	$1.84 \times 10^{-6} \times p(VB\|TO) \times p(race\|VB) = 1.84 \times 10^{-6} \times .83 \times .00012 = 1.83 \times 10^{-10}$	
PPSS	0.02479	0	0	0	
start					
	I	want	to	race	

Answer 2

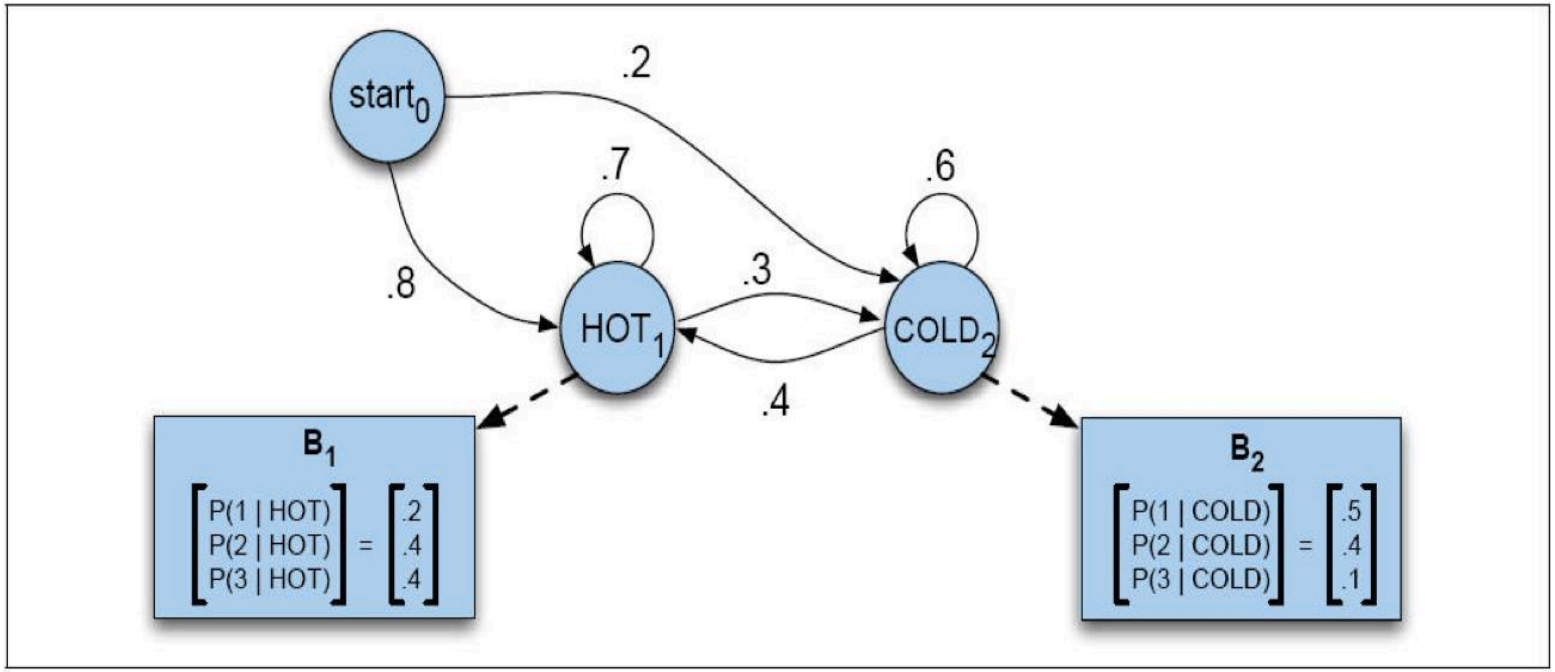
	VB	TO	NN	PPSS
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

	I	want	to	race
VB	0	.0093	0	.00012
TO	0	0	.99	0
NN	0	.000054	0	.00057
PPSS	.37	0	0	0

					end
NN	0	1.6×10^{-9}	0	$1.84 \times 10^{-6} \times p(NN\|TO) \times p(race\|NN) = 1.84 \times 10^{-6} \times .00047 \times .00057 = 4.92 \times 10^{-14}$	
TO	0	0	1.84×10^{-6}	0	
VB	0	5.3×10^{-5}	0	$1.84 \times 10^{-6} \times p(VB\|TO) \times p(race\|VB) = 1.84 \times 10^{-6} \times .83 \times .00012 = 1.83 \times 10^{-10}$	
PPSS	0.02479	0	0	0	
start					
	I	want	to	race	

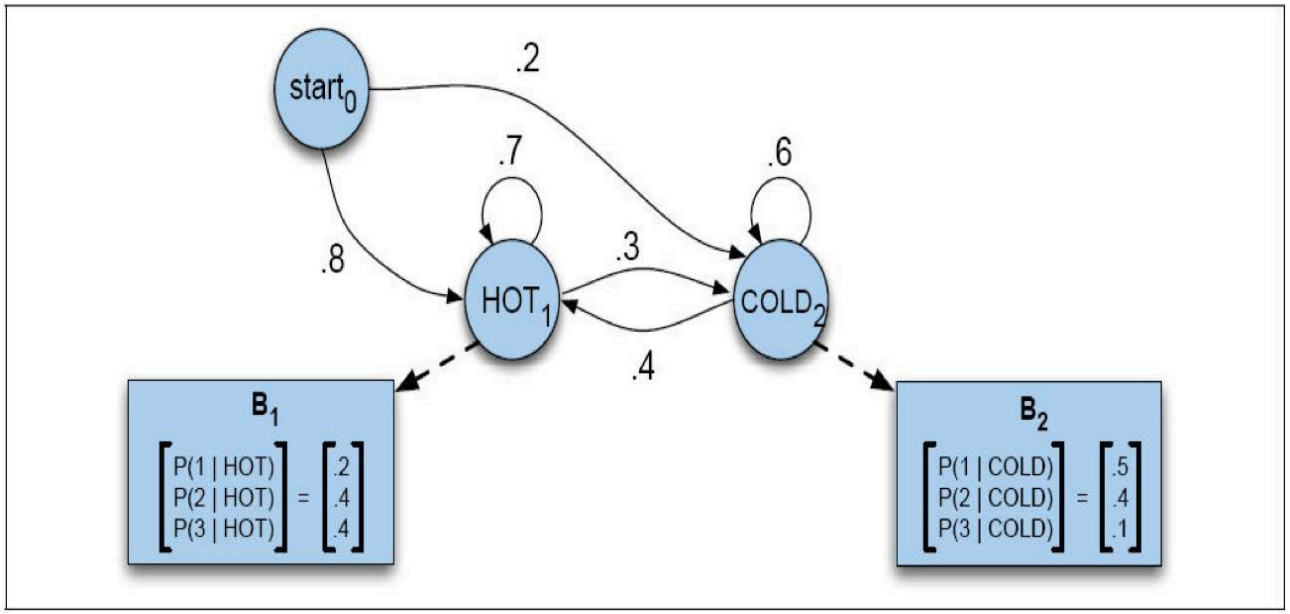
Question 3

- Run the Viterbi algorithm with the HMM below to compute the most likely weather sequences for each of the two observation sequences,
  - 312312312
  - 311233112.

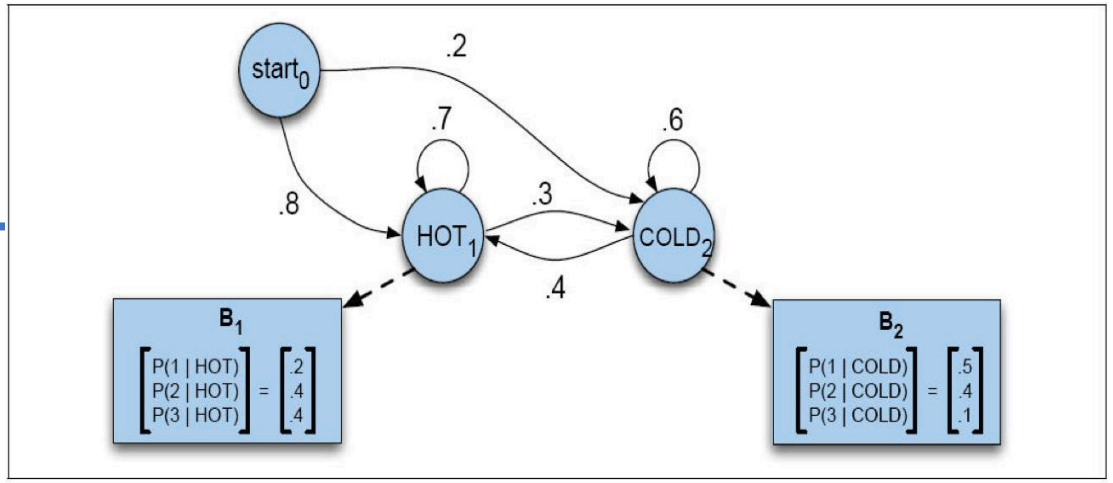


Hint 3

					end
H					
C					
start					
	3	1	2	3	…



Answer 3



- 3
  - H 0.8 *0.4 (P(3|H)) = 0.32
  - C 0.2 *0.1 (P (3|C)) = 0.02
- 1
  - H max (0.32*0.7*0.2, 0.02*0.4*0.2)
  - C max (0.32*0.3*0.5, 0.02*0.6*0.5)
- 2
- 3

https://hmmlearn.readthedocs.io/en/stable/index.html

Sequence 1: 3 1 2 3 1 2 3 1 2
Decoded states: -Hot--Hot--Hot--Hot--Hot--Hot--Hot--Hot--Hot-

Sequence 2: 3 1 1 2 3 3 1 1 2.
Decoded states: -Hot--Cold--Cold--Hot--Hot--Hot--Cold--Cold--Cold-

Question 4

- The task of **negation scope detection** is to extract the parts of a sentence that is being negated.
- For example, in the sentence "I have not submitted my assignment", the negation scope is "submitted my assignment".
- Formulate this problem as a sequence labelling task, and discuss how to apply Hidden Markov Model (HMM) to solve this problem.
- Clearly state the probabilities that need to be learned by the HMM.

Answer 4

- This question is similar to NER, a typical sequence labeling task

- We can use **BIO** label scheme
  - B marks the being of the negation scope.
  - I marks tokens within the negation scope.
  - O marks tokens NOT part of the negation scope.

O	O	O	B	I	I	O
I	have	not	submitted	my	assignment	yet

- In the HMM,
  - the hidden states are BIO (example above), and the word sequence is the observed.
  - We learn three probabilities:
    - transition probability between states BIO,
    - observation likelihood of observing a word given a label B, I, or O.
    - The initial probability of BIO in a sentence.

Question 4 → a bit of extension

- The task of **negation scope detection** is to extract the parts of a sentence that is being negated. For example, in the sentence "I have not submitted my assignment", the negation scope is "submitted my assignment".

- Consider these sentences:

  - He **seldom** makes mistake.
  - I do **not** know why he is **not** happy.
  - He must be very nervous, but he **denied**.
  - He **may or may not** join us for lunch.

Natural Language Processing

Tutorial 5: Grammar and Parsing

Dr. Sun Aixin

Q1. Consider the L1 grammar used in our lectures.

S→ NP VP	Nominal → Noun	NP→ Verb PP
S→ Aux NP VP	Nominal → Nominal Noun	VP → VP PP
S→ VP	Nominal → Nominal PP	PP→ Preposition NP
NP→ Pronoun	VP→ Verb	
NP→ Proper-Noun	VP→ Verb NP	
NP→ Det Nominal	VP → Verb NP PP	

- *Draw the parse table for the following sentence using the L1 grammar*
  - *Reserve a room at MBS*
- Parse table → CKY Parsing

The CKY algorithm

Review

- CKY algorithm requires grammars to be in Chomsky Normal Form (CNF).
  - CNF rules can only be in two forms: $A \rightarrow B\ C$ or $A \rightarrow w$.
  - That is, the right-hand side of each rule must expand either to **two non-terminals** or to **a single terminal**.

- Any CFG can be converted into a corresponding equivalent CNF grammar
  - Rules that mix terminals with non-terminals on the right-hand side
    - e.g., *INF-VP → to VP*. Create a dummy non-terminal *TO*
    - *INF-VP → to VP* becomes *INF-VP → TO VP* and *TO → to*
  - Rules that have a single non-terminal on the right-hand side
    - e.g., *S→ VP*. Rewrite the right-hand side and expand *VP* with all its corresponding rules. *S→ VP* becomes *S→ Verb NP*, *S→ Verb NP PP*, and …
  - Rules that the length of the right-hand side is greater than 2
    - e.g., *S→ Verb NP PP*. Create a dummy non-terminal *X1*. *S→ Verb NP PP* becomes *S→X1 NP*, *X1→Verb NP*

An example CFG grammar in its CNF form

Review

$\mathcal{L}_1$ Grammar	$\mathcal{L}_1$ in CNF
$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

CNF rules can only be in two forms: ***A → B C*** or ***A → w***.

Each non-terminal node above the POS level in a parse tree will have **exactly two** daughters

That is: a non-terminal node can be derived from **exactly TWO constituents** (that can be derived earlier).

An example CFG grammar in its CNF form

$\mathcal{L}_1$ Grammar	$\mathcal{L}_1$ in CNF
$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

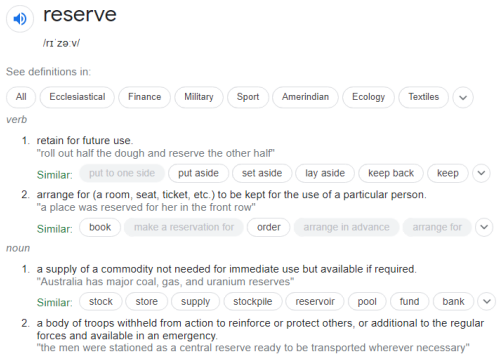
|reserve

|MBS

|reserve

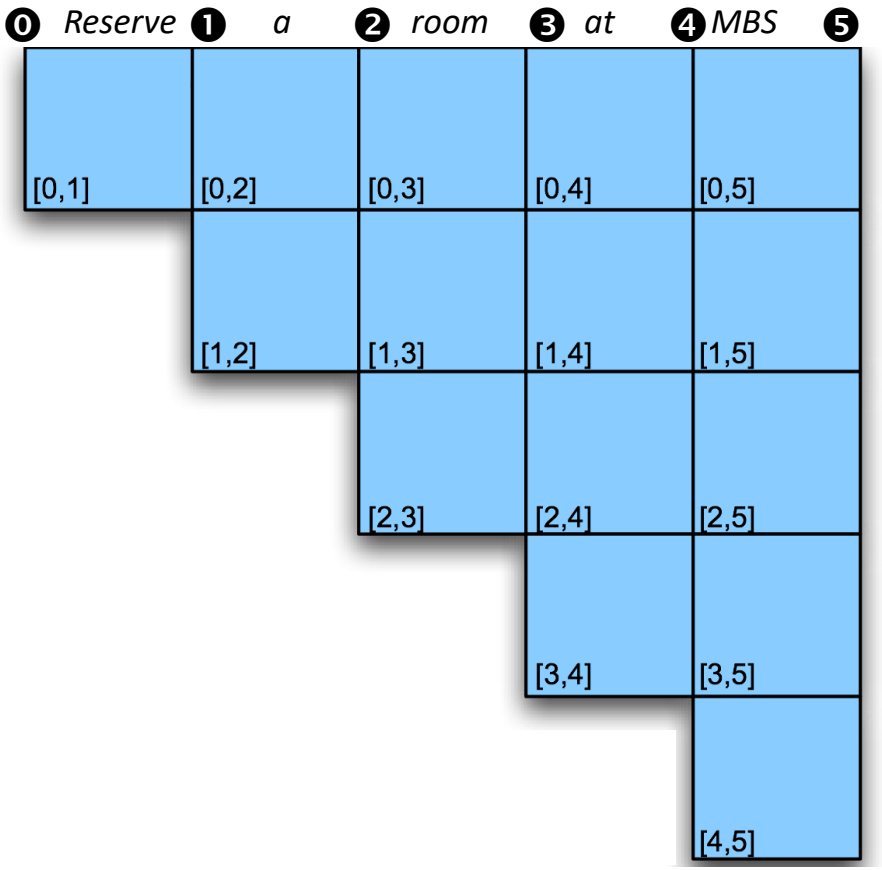
Reserve a room at MBS

reserve|room



Q2(i)

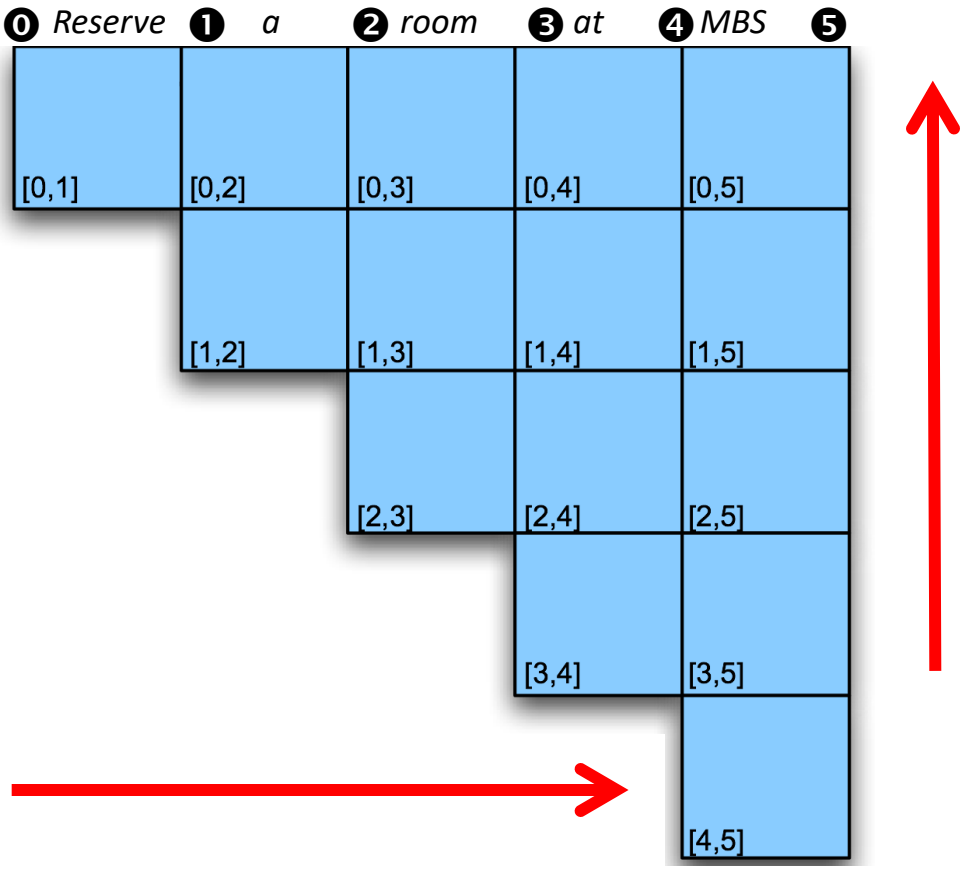
$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



"reserve" can be a noun or verb, similar to "book"

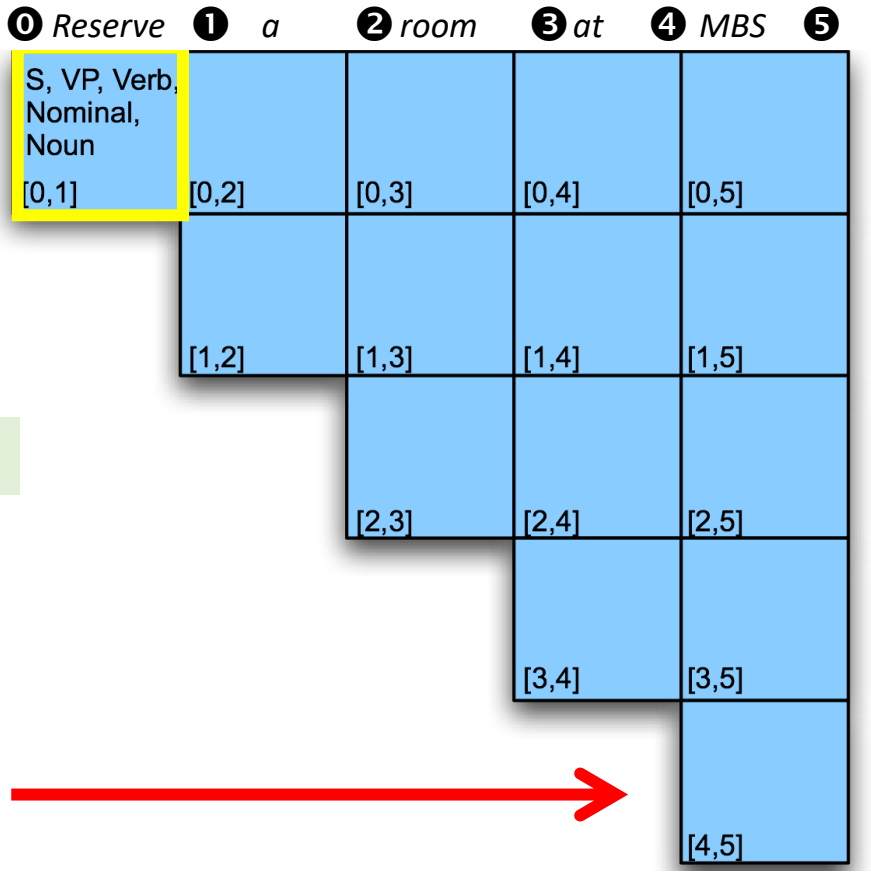
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



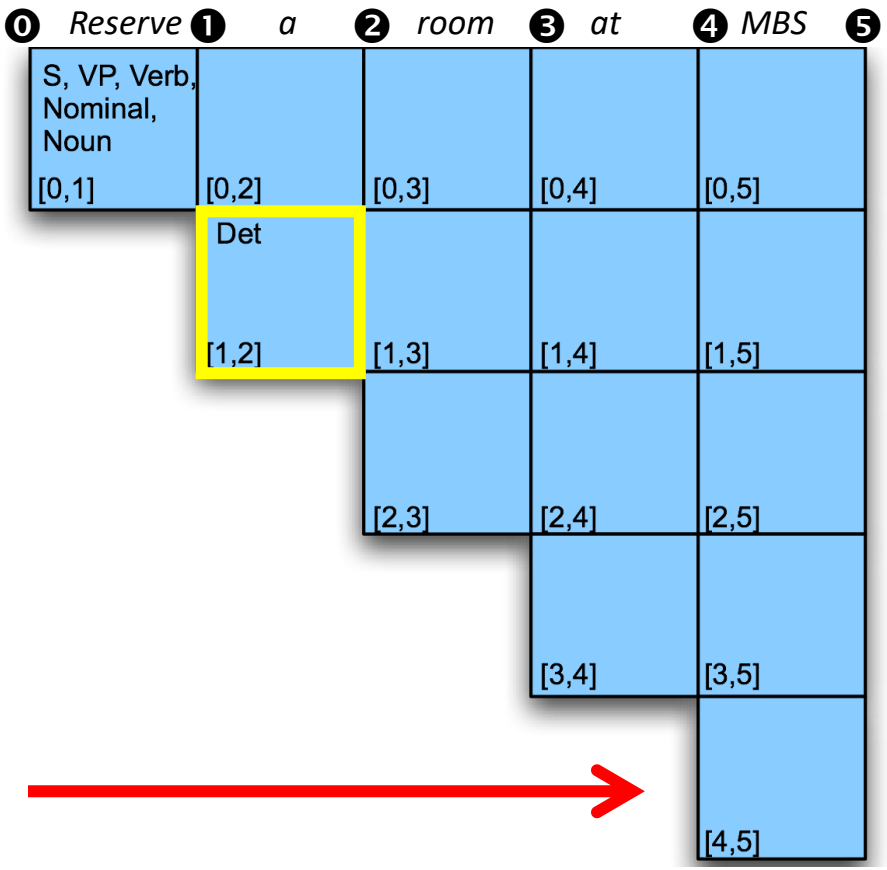
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$ |reserve
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$ |reserve
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$ |reserve
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



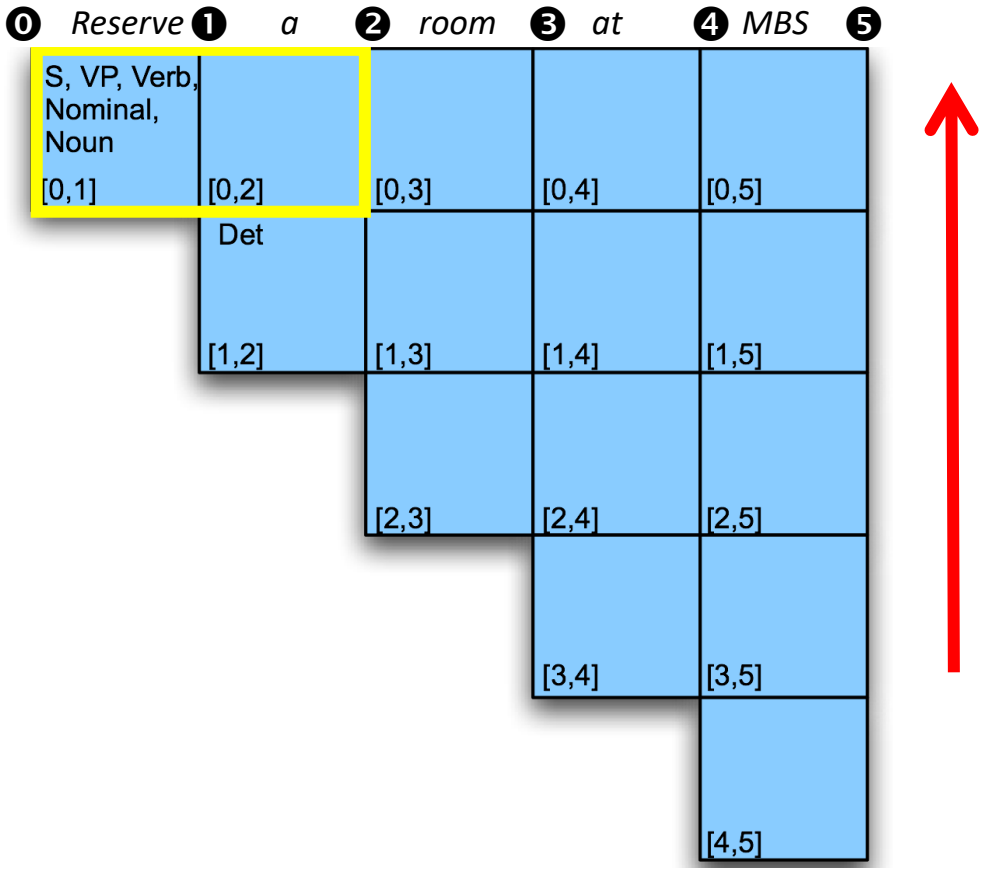
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



Q2(i)

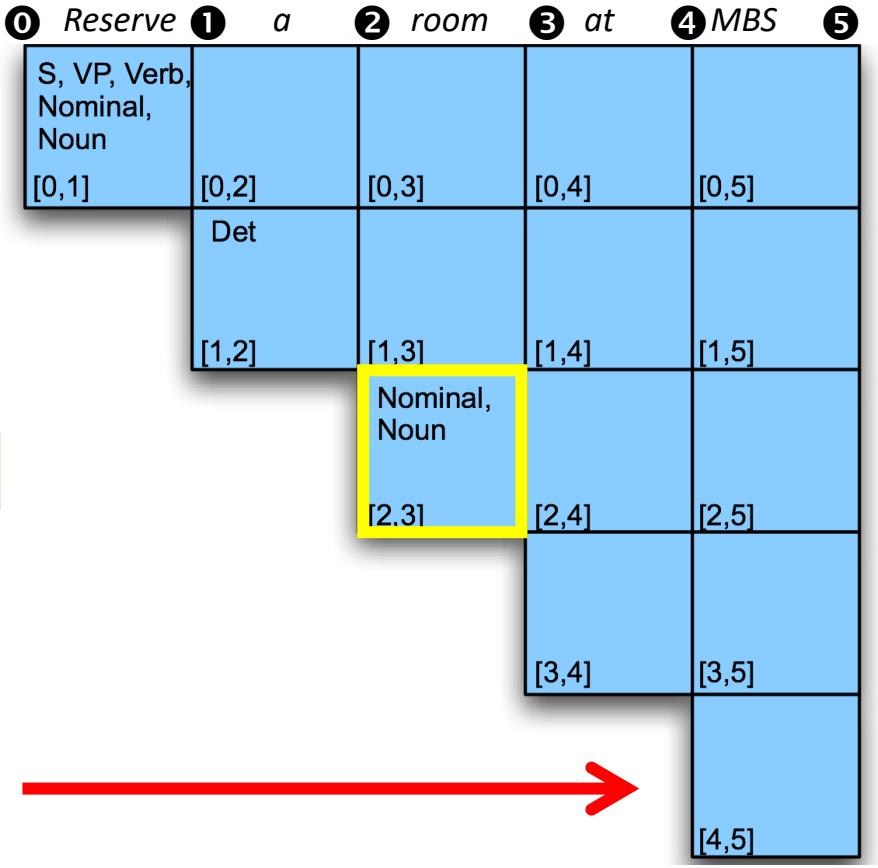
$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



Q2(i)

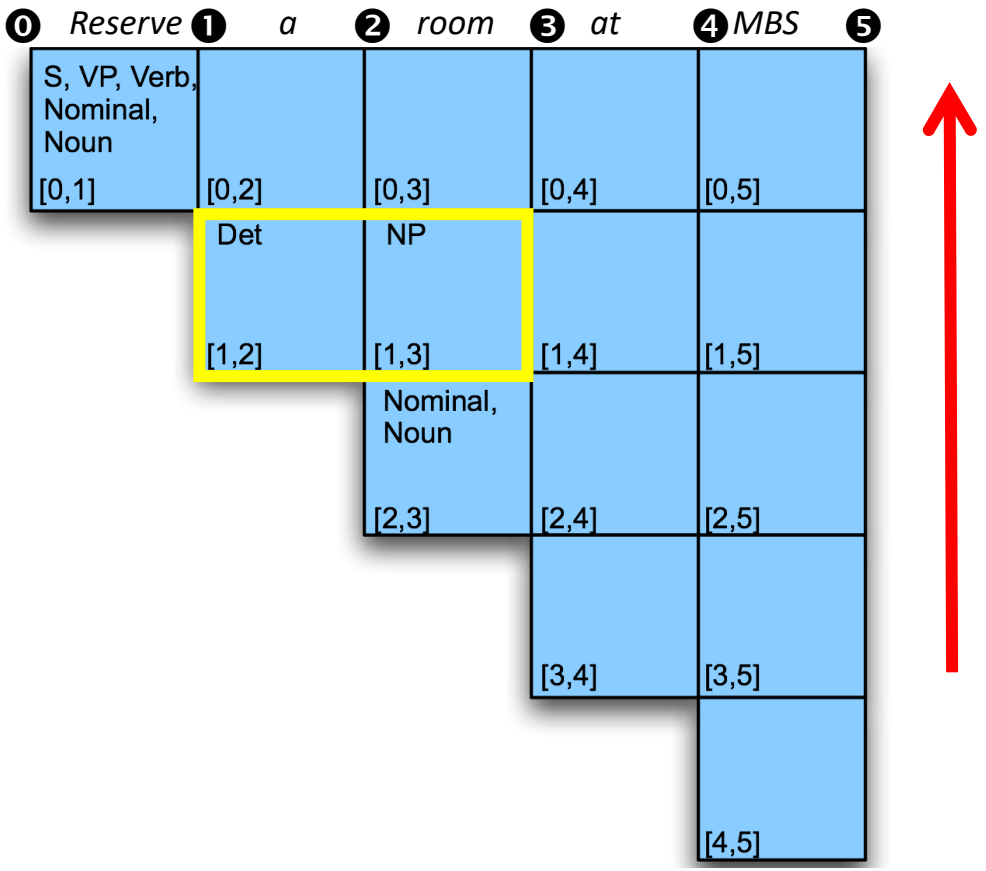
$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$

|room



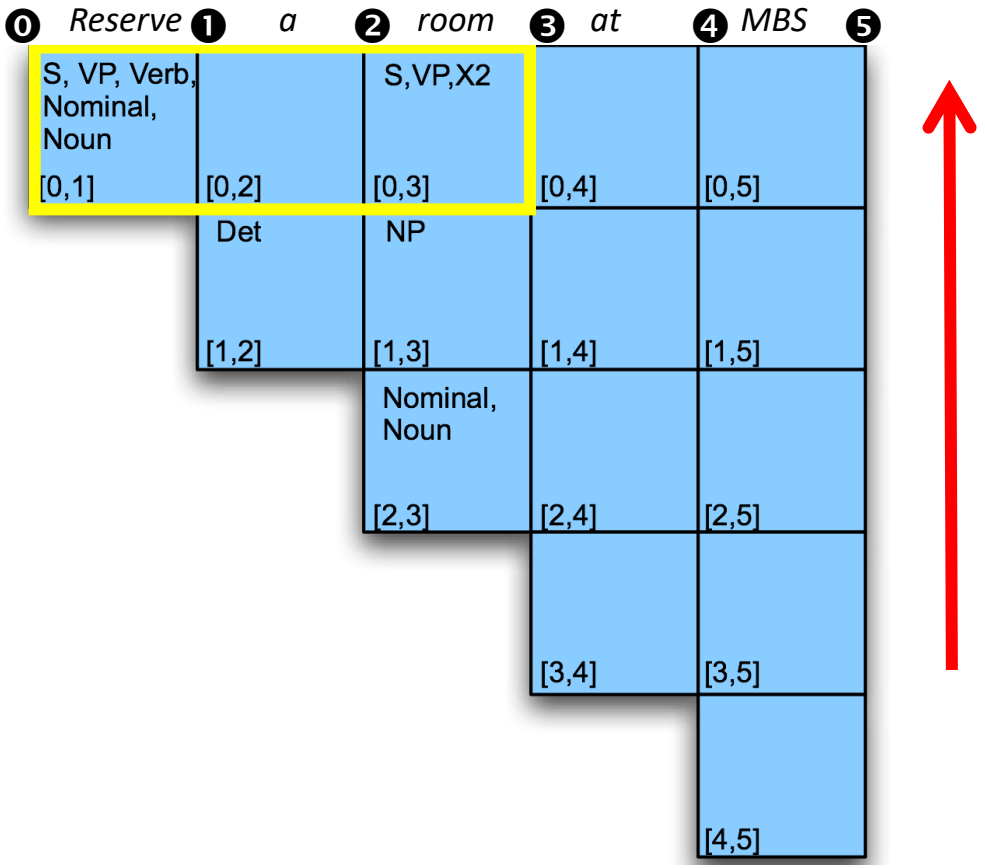
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



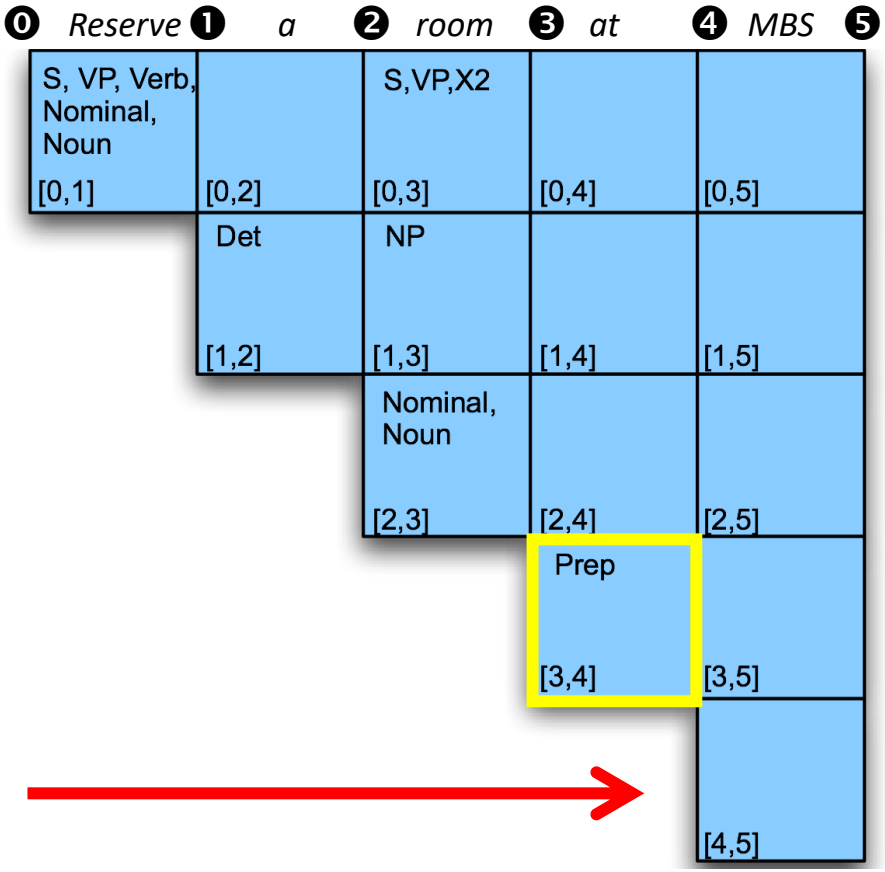
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



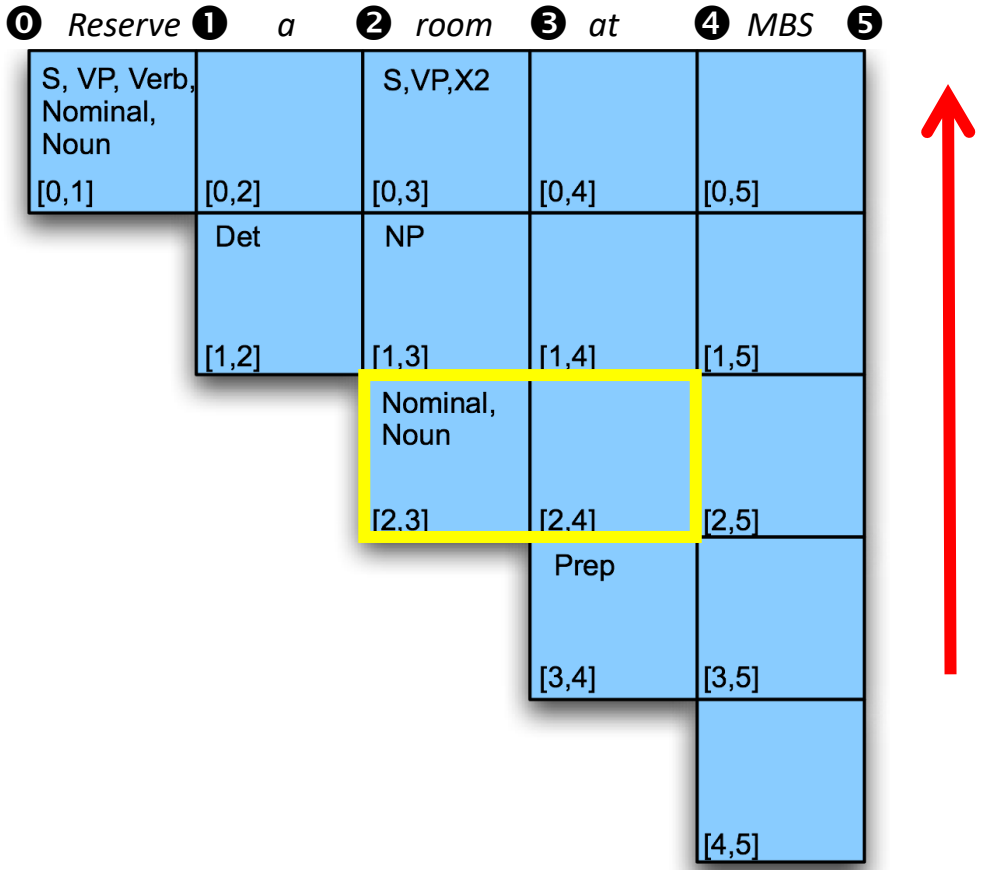
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



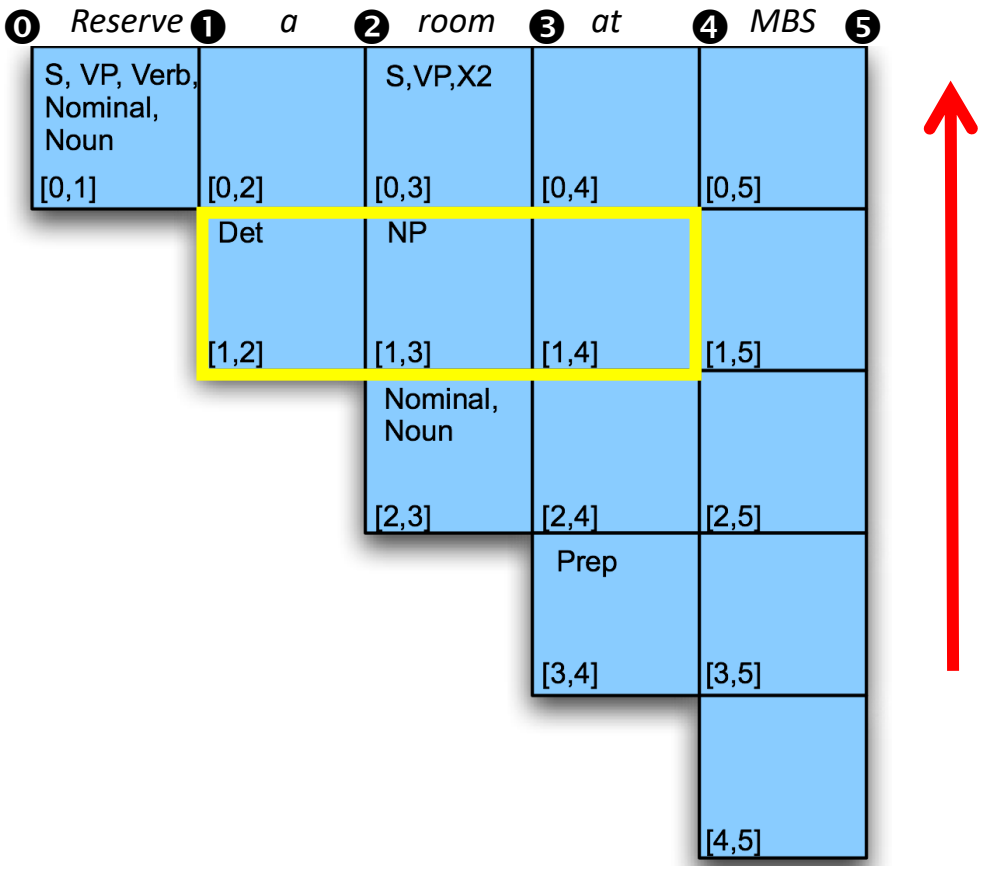
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



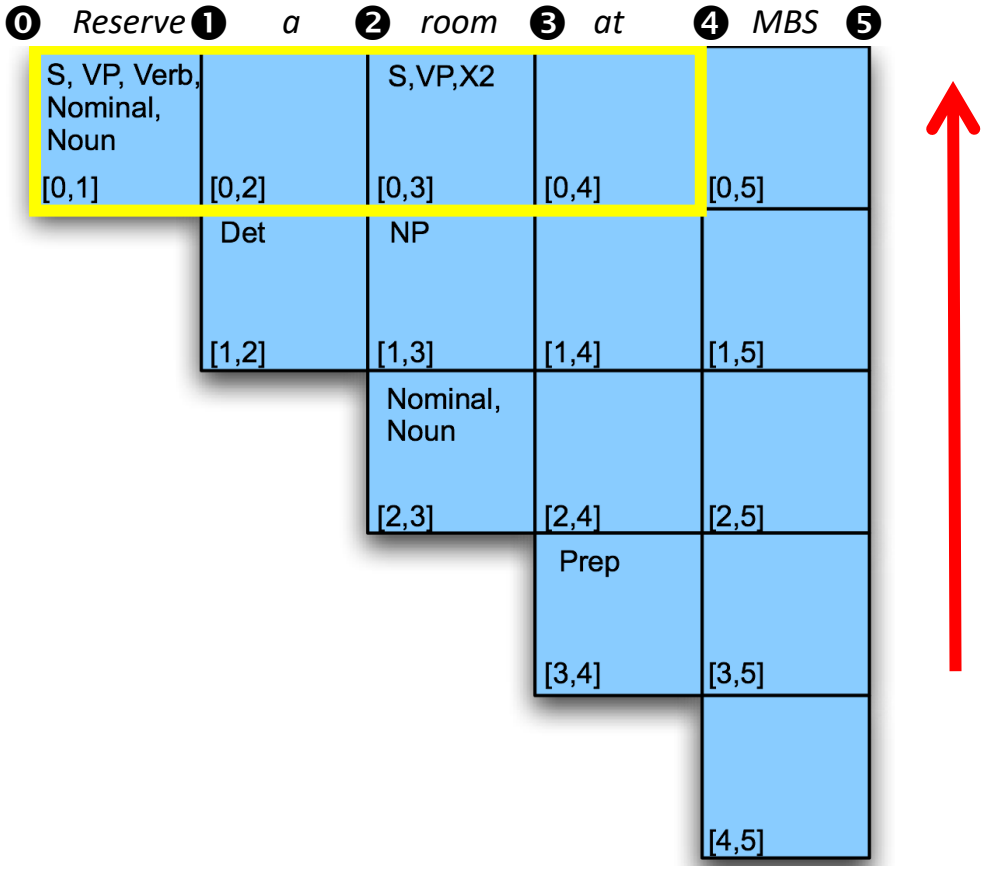
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



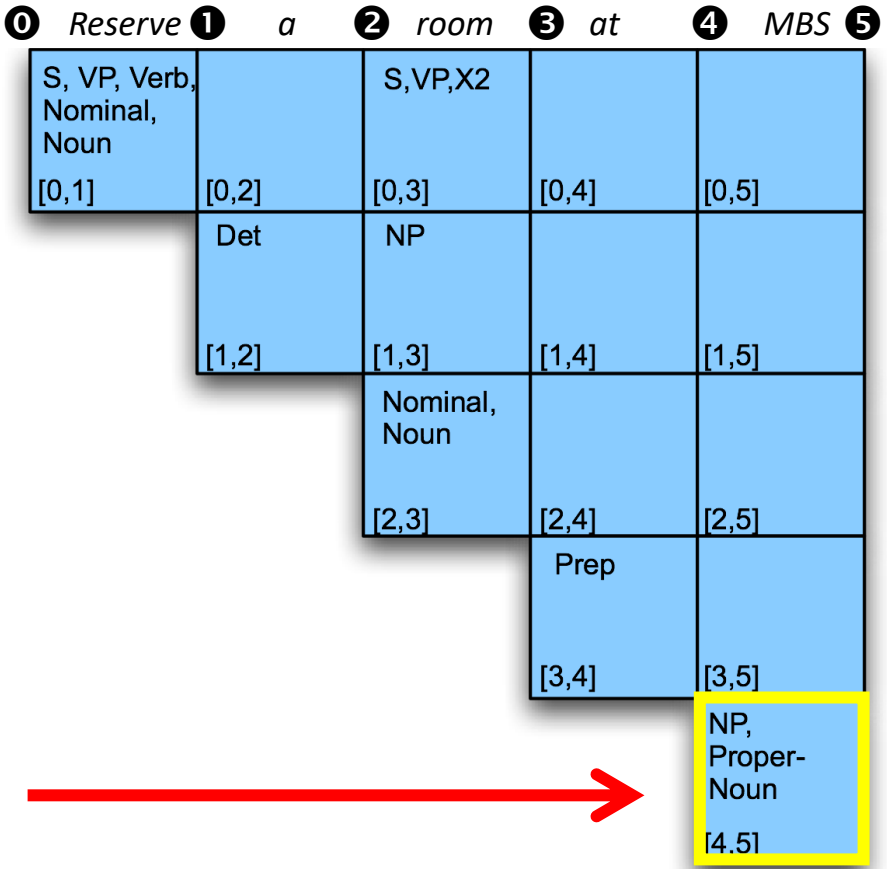
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



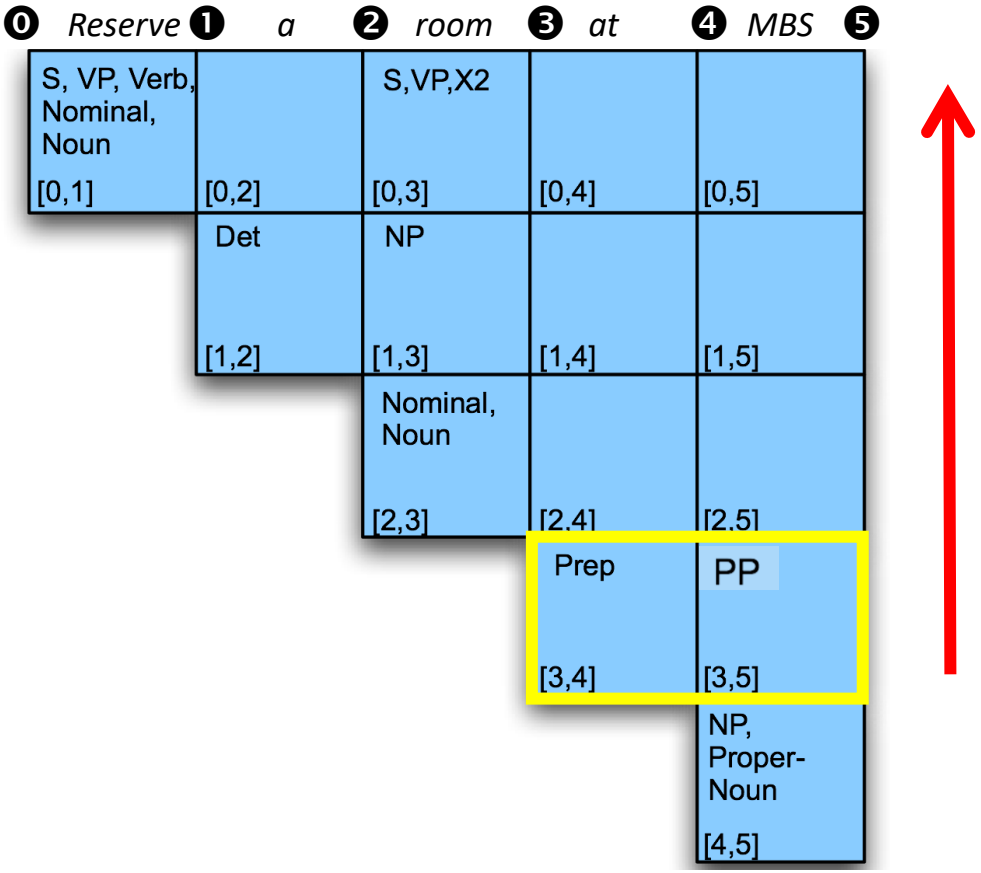
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



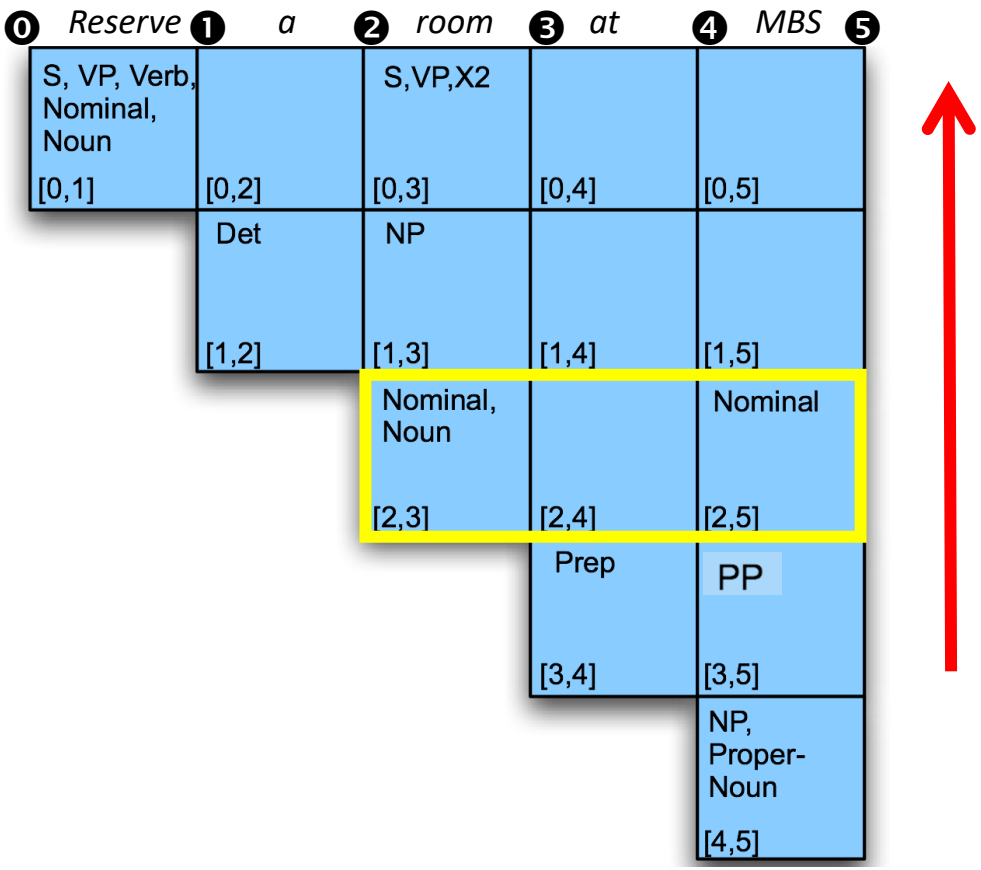
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



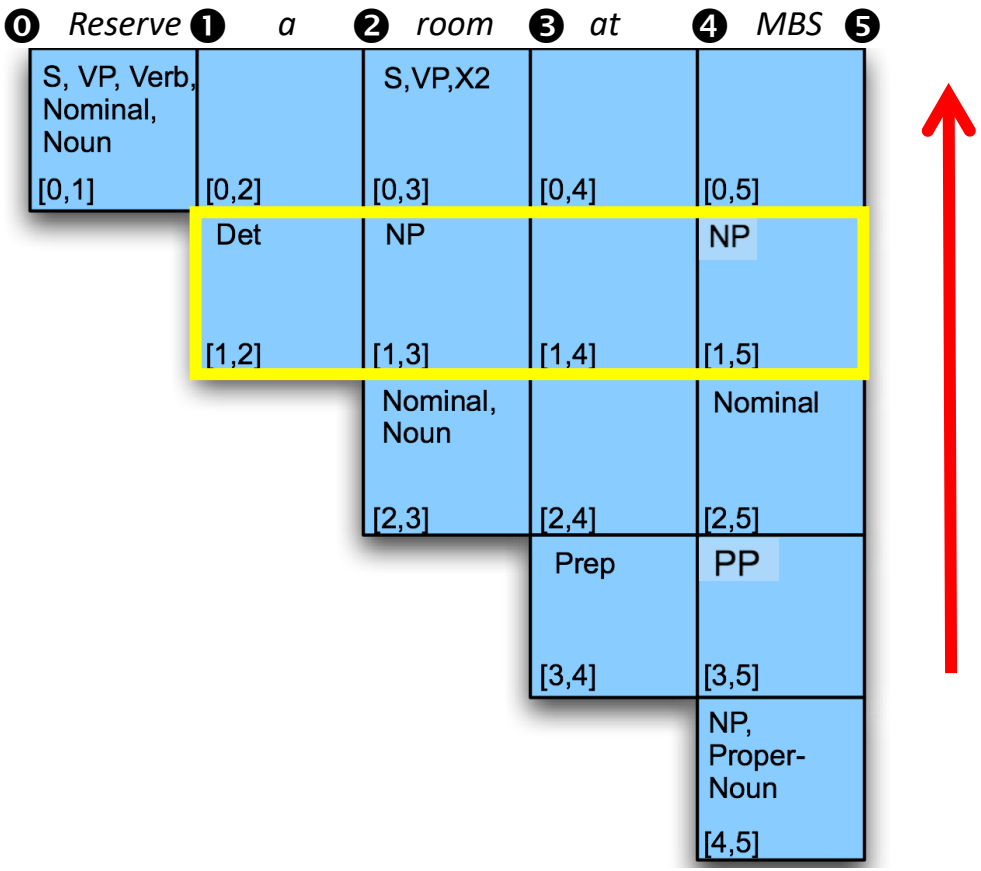
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



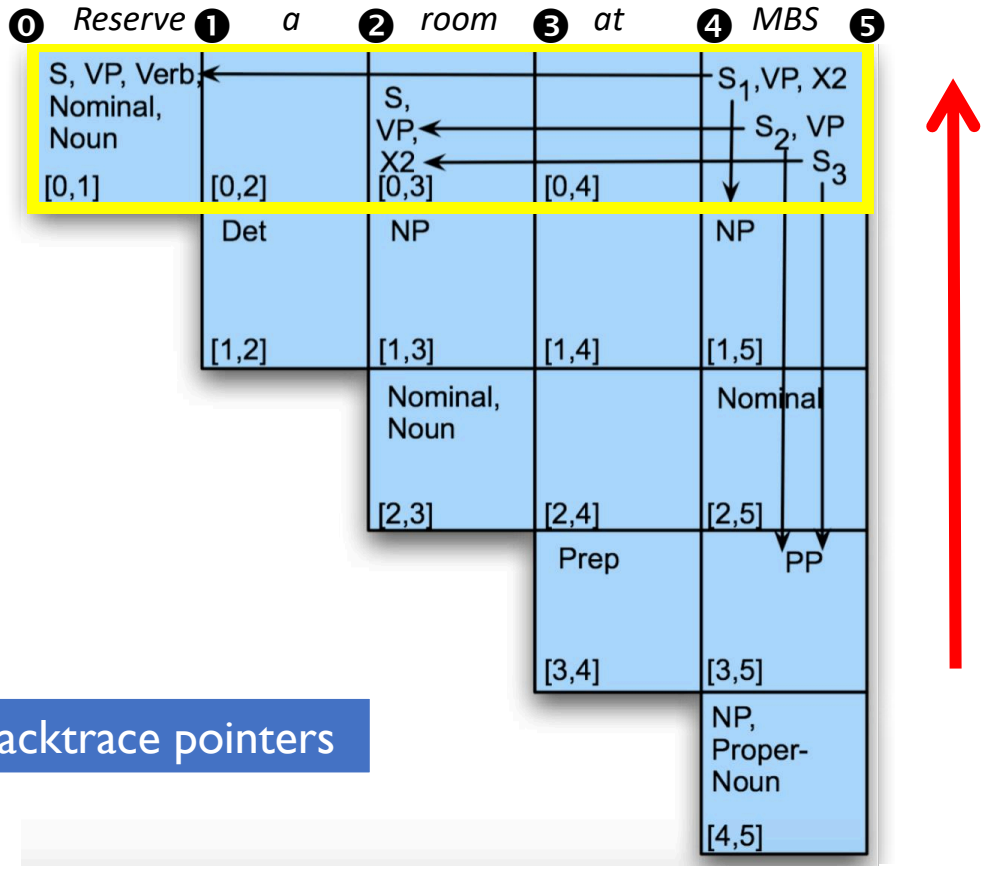
Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



Q2(i)

$\mathscr{L}_1$ in CNF
$S \rightarrow NP\ VP$
$S \rightarrow X1\ VP$
$X1 \rightarrow Aux\ NP$
$S \rightarrow book \mid include \mid prefer$
$S \rightarrow Verb\ NP$
$S \rightarrow X2\ PP$
$S \rightarrow Verb\ PP$
$S \rightarrow VP\ PP$
$NP \rightarrow I \mid she \mid me$
$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$
$VP \rightarrow X2\ PP$
$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$



Backtrace pointers

Q2 You are provided with the phrase structures for the following sentences:

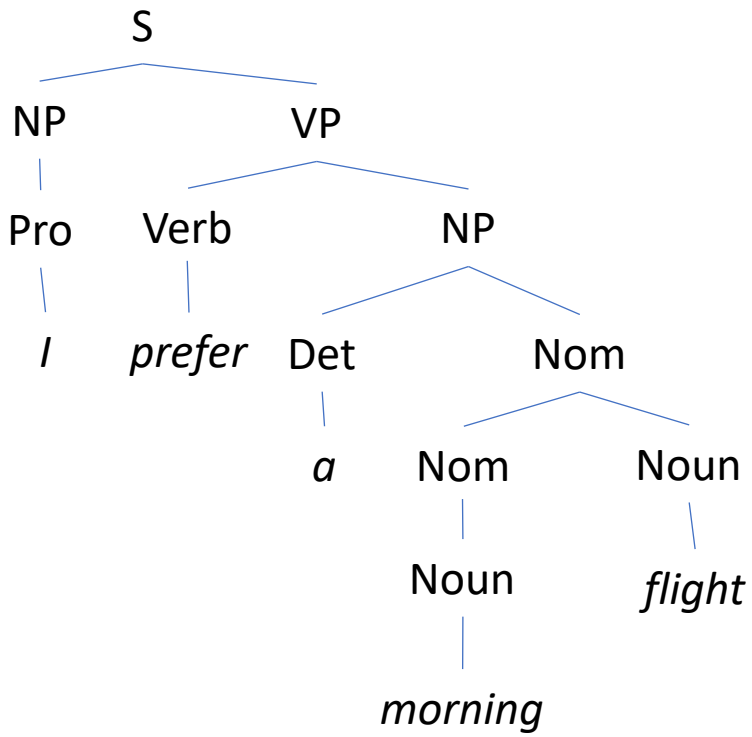
- Sentences:
  - I would like to fly on American airlines.
  - Please repeat that.
  - I need to fly between Philadelphia and Atlanta.
  - What is the fare from Atlanta to Denver?
- **Revise the L1 grammar Q1 such that the revised grammar can be used to parse the above four sentences**

S→ NP VP	Nominal → Noun	NP→ Verb PP
S→ Aux NP VP	Nominal → Nominal Noun	VP → VP PP
S→ VP	Nominal → Nominal PP	PP→ Preposition NP
NP→ Pronoun	VP→ Verb	
NP→ Proper-Noun	VP→ Verb NP	
NP→ Det Nominal	VP → Verb NP PP	

Derivation

Review

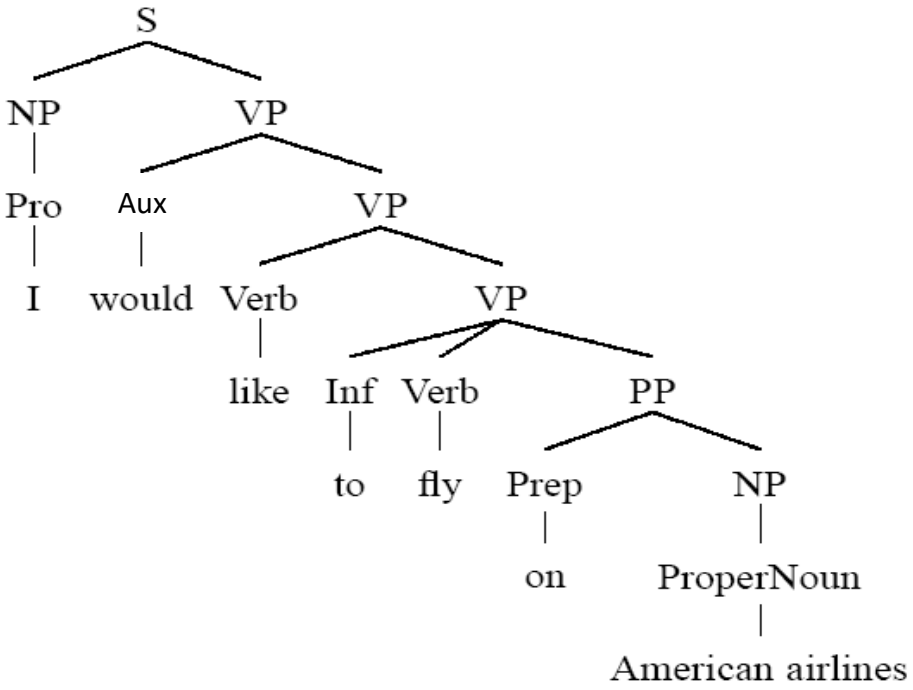
- A derivation (parse tree) consists of the bag of grammar rules that are in the tree

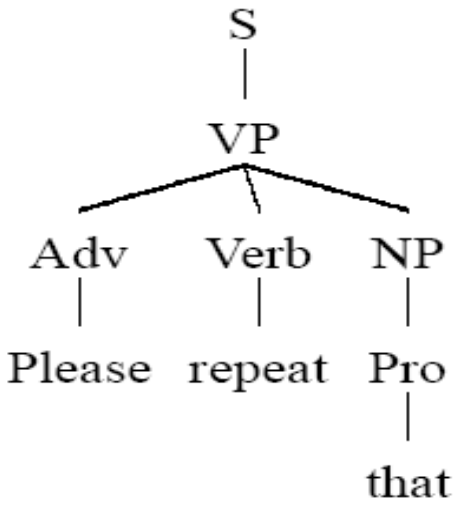


1. S → NP VP
2. NP → Pro
   Pro → I
3. VP → Verb NP
   Verb → prefer
4. NP → Det Nom
   Det → a
5. Nom → Nom Noun
   Noun → morning
6. Nom → Noun
   Noun → flight

The first two sentences

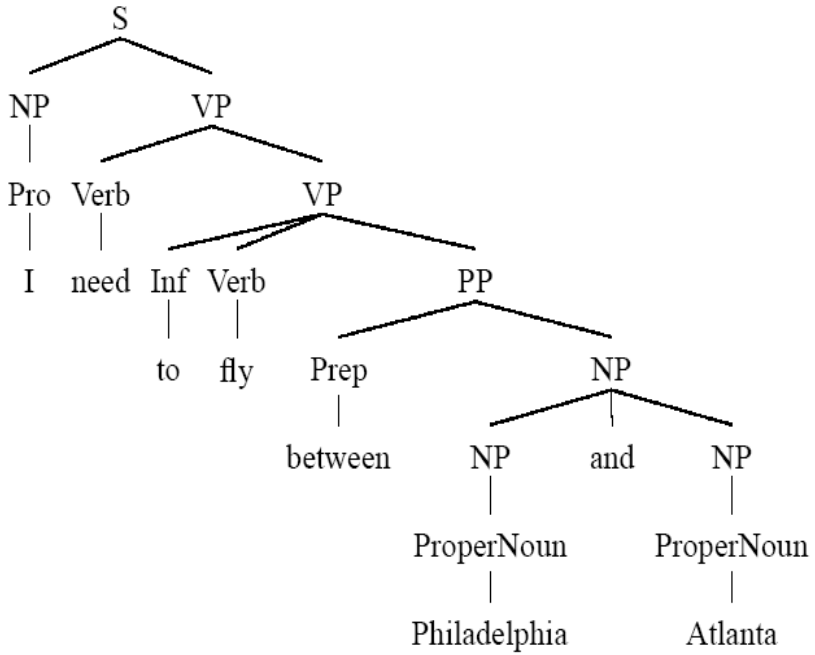
- I would like to fly on American airlines.
- Please repeat that.

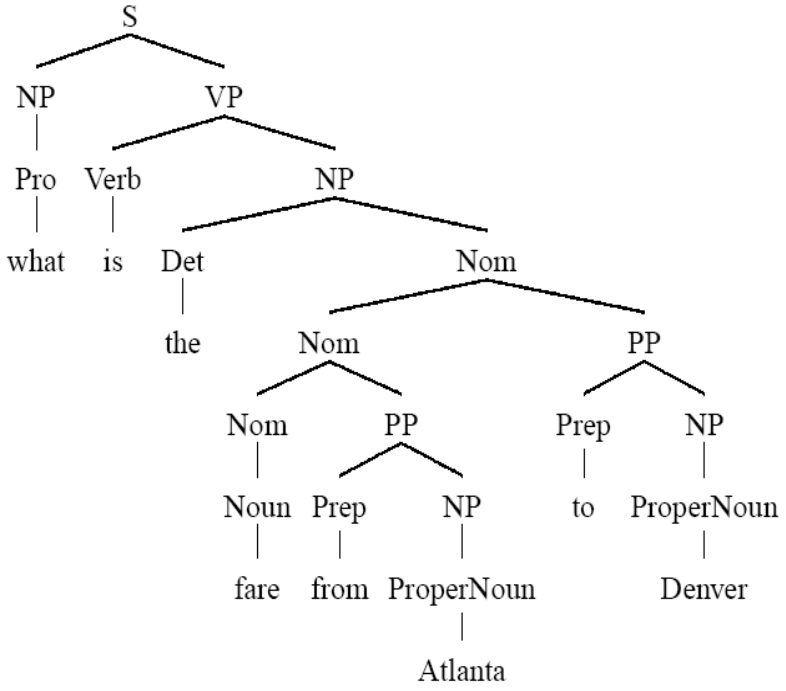




The next two sentences

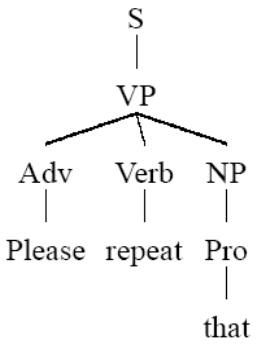
- I need to fly between Philadelphia and Atlanta.
- What is the fare from Atlanta to Denver?





Q2 Solution: Include the rules that are not listed in L1

- S → NP VP
- S → Aux NP VP
- S → VP
- NP → Pronoun
- NP → ProperNoun
- NP → Det Nominal
- NP → NP Conj NP
- Nominal → Noun
- Nominal → Nominal Noun
- Nominal → Nominal PP
- VP → Verb
- VP → Verb NP
- VP → Verb NP PP
- VP → Verb PP



- VP → VP PP
- VP → Aux VP
- VP → Verb VP
- VP → Inf Verb PP
- VP → Adv Verb NP
- PP → Preposition NP
- ----------
- Det → the
- Noun → fare
- Verb → like | fly | repeat | need | is
- Pronoun → I | that | what
- ProperNoun → American airlines | Philadelphia | Atlanta | Denver
- Aux → would
- Preposition → from | to | on | between
- Conj → and
- Inf → to
- Adv → please

Q3. Use the following sentences as examples to observe the dependency structures

https://demos.explosion.ai/displacy

a) Does American airlines have a flight between five a.m. and six a.m.?

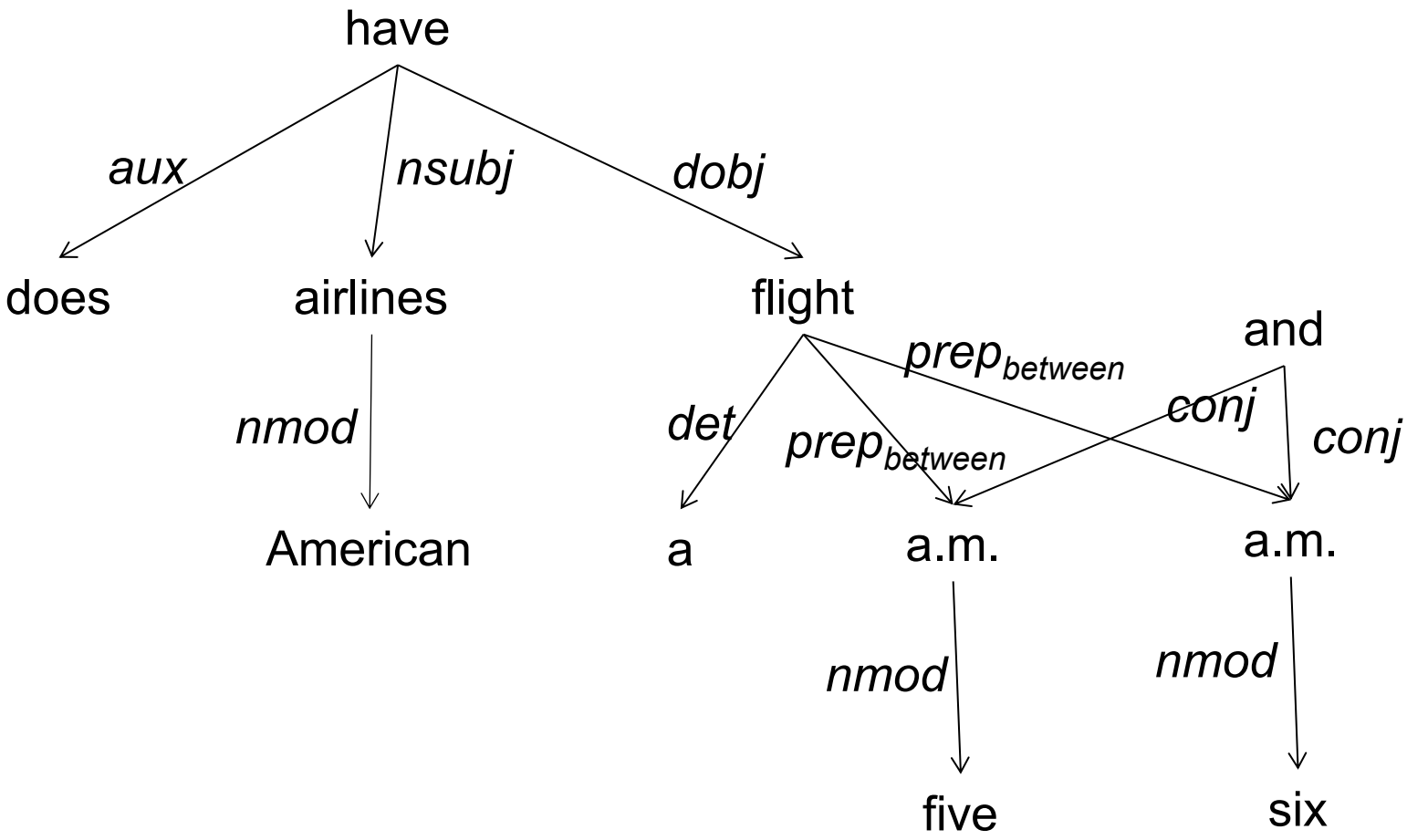
b) I would like to fly on American airlines.

c) Please repeat that.

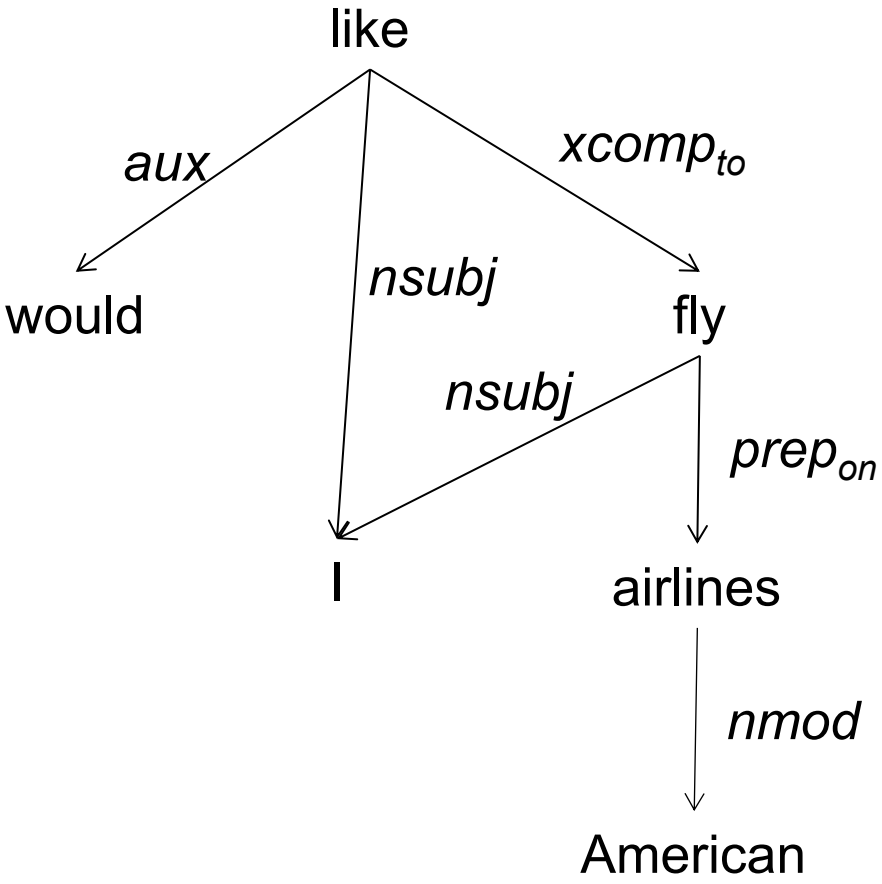
d) I need to fly between Philadelphia and Atlanta.

e) What is the fare from Atlanta to Denver?

Q3 a). Does American airlines have a flight between five a.m. and six a.m.?

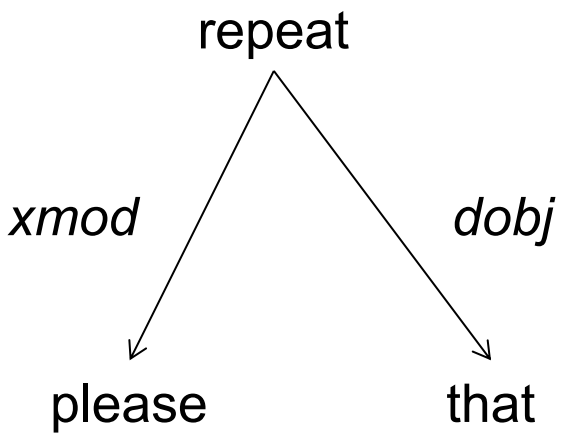


Q3 b). I would like to fly on American airlines.



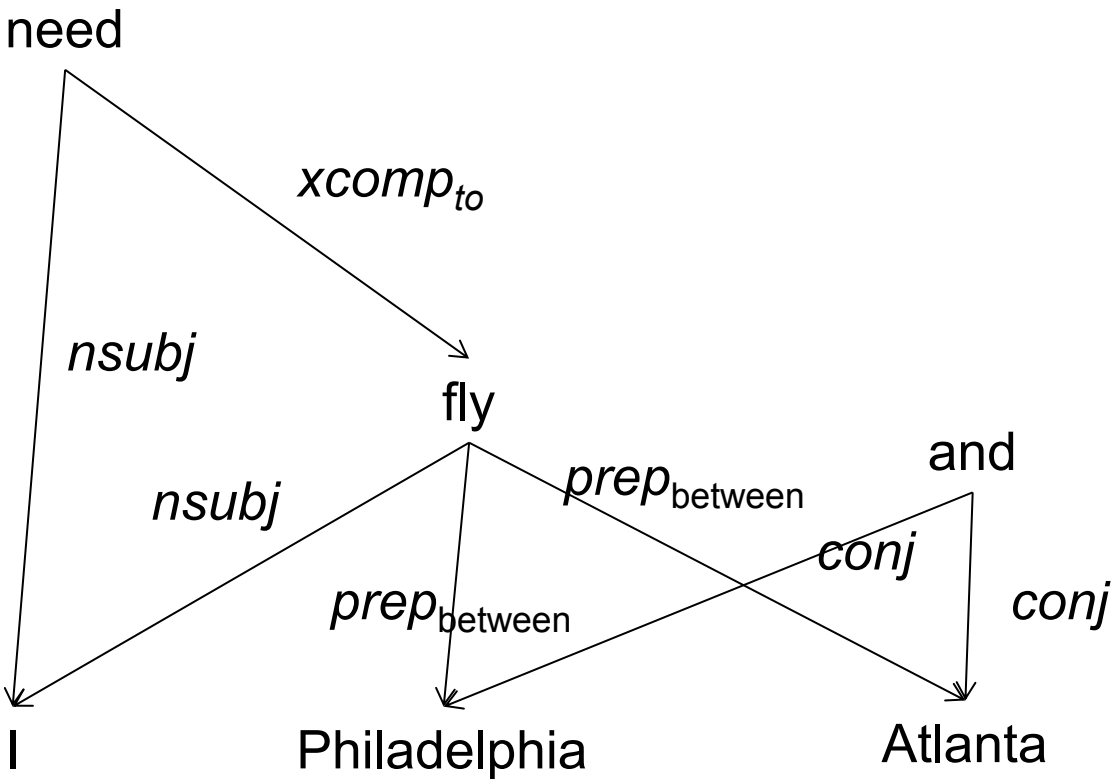
- Verb relations
  - nsubj
  - dobj
  - aux (auxiliary verb – main verb)
  - xmod (verb – adverb)
  - $xcomp_{to}$ (verb – to-infinitive)

Q3 c). Please repeat that.



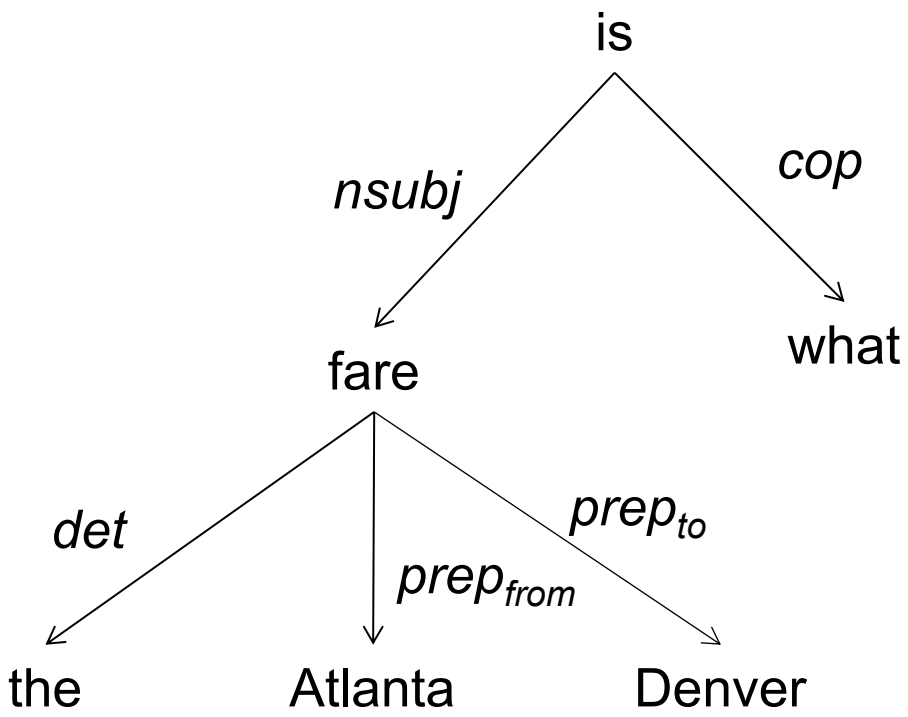
- Verb relations
  - nsubj
  - dobj
  - aux (auxiliary verb – main verb)
  - xmod (verb – adverb)
  - $xcomp_{to}$ (verb – to-infinitive)

Q3 d). I need to fly between Philadelphia and Atlanta.



- Verb relations
  - nsubj
  - dobj
  - aux (auxiliary verb – main verb)
  - xmod (verb – adverb)
  - $xcomp_{to}$ (verb – to-infinitive)

Q3 e). What is the fare from Atlanta to Denver?



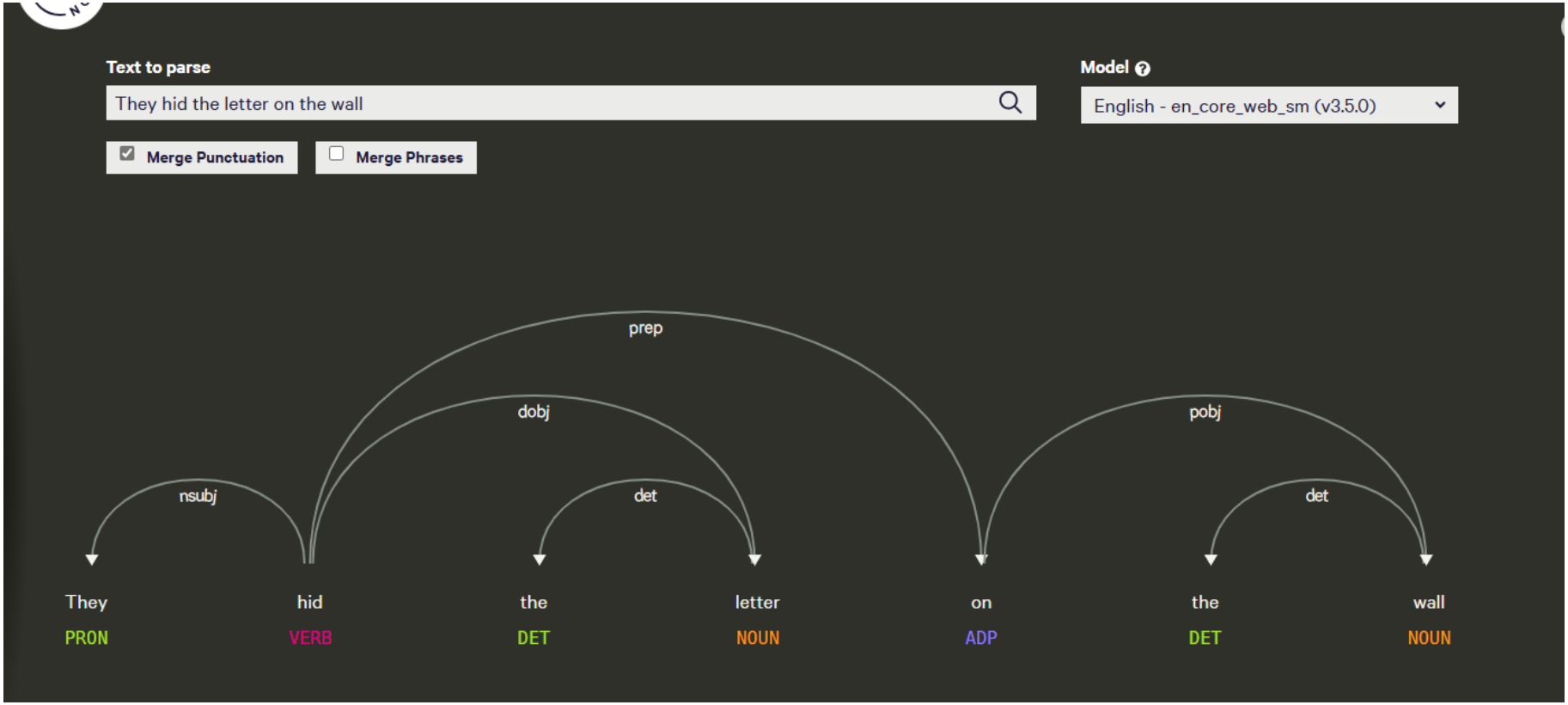
- Verb relations
  - nsubj
  - dobj
  - aux (auxiliary verb – main verb)
  - xmod (verb – adverb)
  - $xcomp_{to}$ (verb – to-infinitive)

Q4 Draw two dependency structures for the following sentence: *They hid the letter on the wall.*

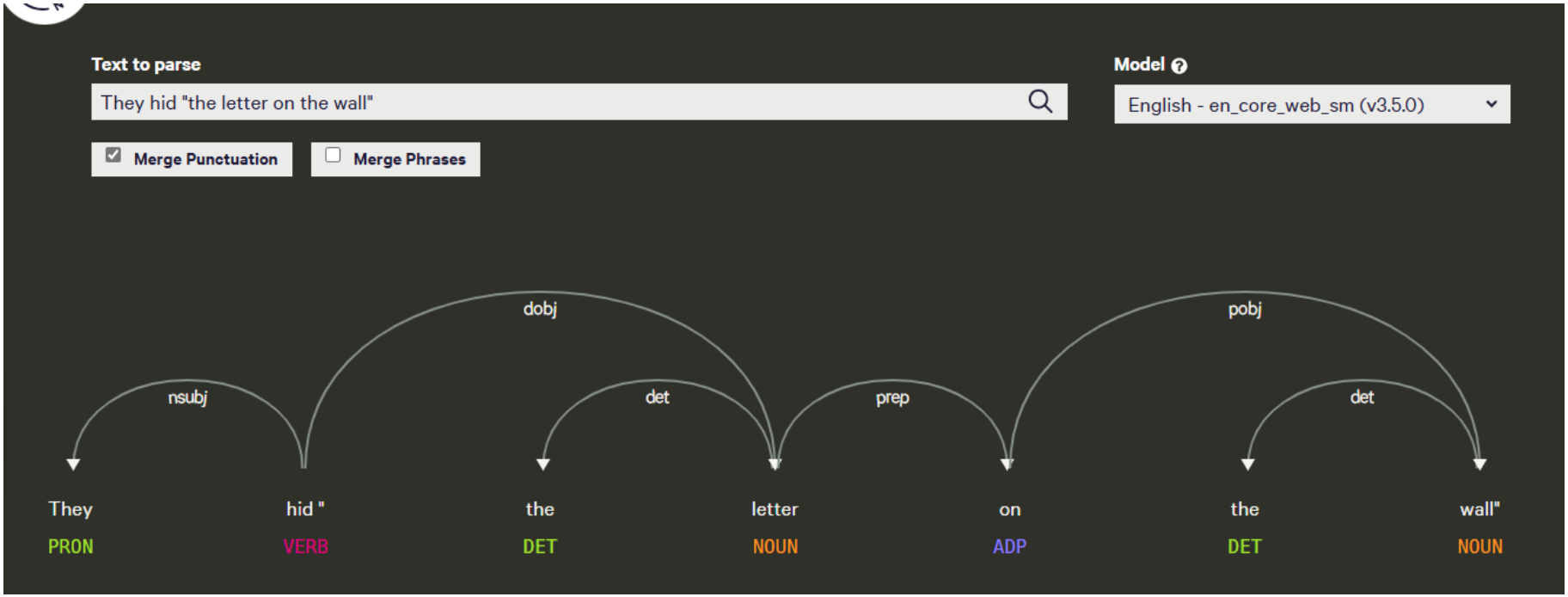
- There was a letter on the wall, but they hid it, and now I've searched the entire wall and can't find it anymore.

- There was a letter on the table, and they hid it on the wall, like behind a painting

Structure I


Text to parse
They hid the letter on the wall
Model
English - en_core_web_sm (v3.5.0)
Merge Punctuation
Merge Phrases
prep
dobj
pobj
nsubj
det
det
They
PRON
hid
VERB
the
DET
letter
NOUN
on
ADP
the
DET
wall
NOUN

Structure 2

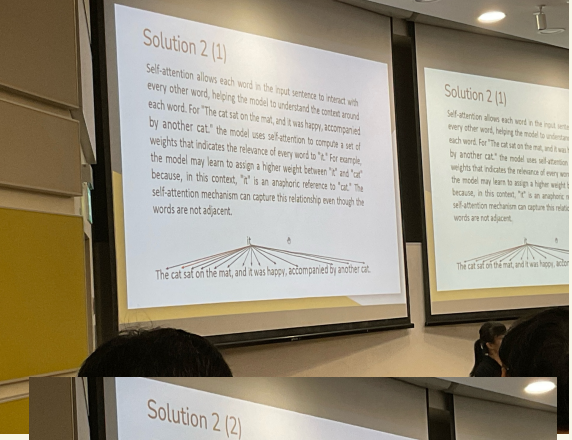

Text to parse
They hid "the letter on the wall"
Model
English - en_core_web_sm (v3.5.0)
Merge Punctuation
Merge Phrases
nsubj
dobj
det
prep
pobj
det
They
PRON
hid "
VERB
the
DET
letter
NOUN
on
ADP
the
DET
wall"
NOUN

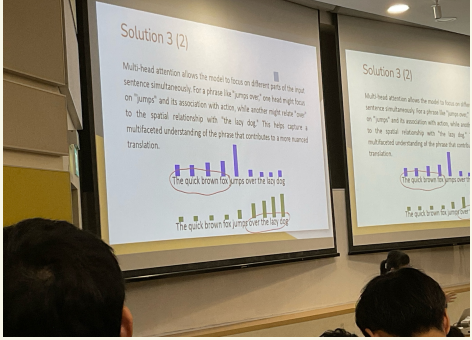
Q5.Open question

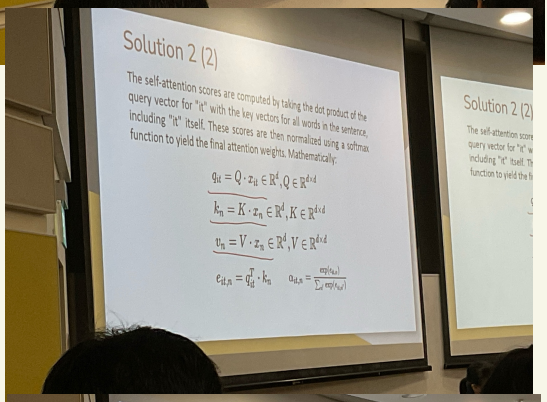
- Try and explore the Rule-based Matcher

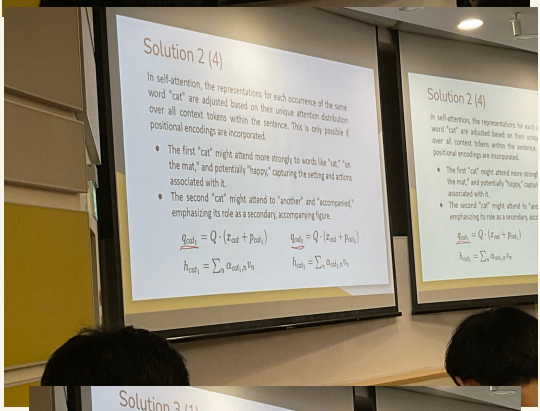
  https://demos.explosion.ai/matcher

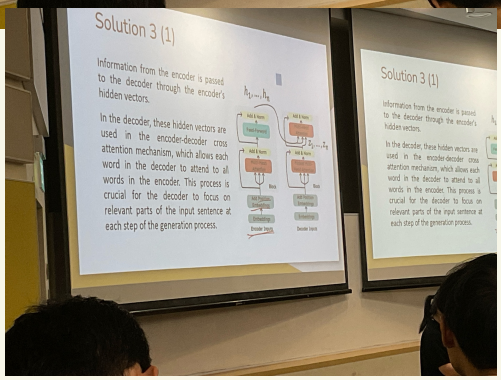
- Define a pattern and explain why it could be useful for a real-life use case for some text data (e.g., news articles, financial reports, medical reports, product reviews)


Solution 2 (1)
Self-attention allows each word in the input sentence to interact with every other word, helping the model to understand the context around each word. For "The cat sat on the mat, and it was happy, accompanied by another cat," the model uses self-attention to compute a set of weights that indicates the relevance of every word to "it." For example, the model may learn to assign a higher weight between "it" and "cat" because, in this context, "it" is an anaphoric reference to "cat." The self-attention mechanism can capture this relationship even though the words are not adjacent.
The cat sat on the mat, and it was happy, accompanied by another cat.
Solution 2 (1)


Solution 3 (2)
Multi-head attention allows the model to focus on different parts of the input sentence simultaneously. For a phrase like "jumps over," one head might focus on "jumps" and its association with action, while another might relate "over" to the spatial relationship with "the lazy dog." This helps capture a multifaceted understanding of the phrase that contributes to a more nuanced translation.
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
Solution 3 (2)


Solution 2 (2)
The self-attention scores are computed by taking the dot product of the query vector for "it" with the key vectors for all words in the sentence, including "it" itself. These scores are then normalized using a softmax function to yield the final attention weights. Mathematically:
$q_{it} = Q \cdot x_{it} \in \mathbb{R}^d, Q \in \mathbb{R}^{d \times d}$
$k_n = K \cdot x_n \in \mathbb{R}^d, K \in \mathbb{R}^{d \times d}$
$v_n = V \cdot x_n \in \mathbb{R}^d, V \in \mathbb{R}^{d \times d}$
$e_{it,n} = q_{it}^T \cdot k_n \quad a_{it,n} = \frac{\exp(e_{it,n})}{\sum_i \exp(e_{it,n})}$
Solution 2 (2)


Solution 2 (4)
In self-attention, the representations for each occurrence of the same word "cat" are adjusted based on their unique attention distribution over all context tokens within the sentence. This is only possible if positional encodings are incorporated.
The first "cat" might attend more strongly to words like "sat," "on the mat," and potentially "happy," capturing the setting and actions associated with it.
The second "cat" might attend to "another" and "accompanied," emphasizing its role as a secondary, accompanying figure.
$q_{cat_1} = Q \cdot (x_{cat} + p_{cat_1})$ $q_{cat_2} = Q \cdot (x_{cat} + p_{cat_2})$
$h_{cat_1} = \sum_n a_{cat_1,n} v_n$ $h_{cat_2} = \sum_n a_{cat_2,n} v_n$
Solution 2 (4)


Solution 3 (1)
Information from the encoder is passed to the decoder through the encoder's hidden vectors.
In the decoder, these hidden vectors are used in the encoder-decoder cross attention mechanism, which allows each word in the decoder to attend to all words in the encoder. This process is crucial for the decoder to focus on relevant parts of the input sentence at each step of the generation process.
$h_1, \ldots, h_n$
$s_1, \ldots, s_m$
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Positional Encoding
Embedding
Encoder Inputs
Positional Encoding
Embedding
Decoder Inputs
Solution 3 (1)

TUT : Recommendation system

Q1

User based CF of 3 most similar users.

Cosine similarity of [u1, u3..u12] with u2:

[0.372, 0.29, 0.217, 0.527, 0.0, 0.325, 0.198, 0.475, 0.667, 0.487, 0.0]

Rankings of users based on similarity:

[10, 5, 11, 9, 1, 7, 3, 4, 8, 12, 6]

Top 3 users who rated movie 1:

u11, u9, u1 (because u10 and u5 didn't rate movie 1)

Similarity-weighted recommendation:

$$\frac{0.487 * 4 + 0.475 * 5 + 0.372 * 1}{0.487 + 0.475 + 0.372} = 3.519$$

Unweighted recommendation:

$$\frac{4 + 5 + 1}{3} = 3.33$$

Item based CF of 3 most similar items.

Step 1:

Cosine similarity of [i2...i12] with i1:

[0.528, 0.526, 0.285, 0.302, 0.239, 0.470, 0.913, 0.681, 0.533, 0.257, 0.465]

Rankings of items based on similarity:

[8, 9, 10, 2, 3, 7, 12, 5, 4, 11, 6]

Top 3 items that interact with u2:

i10, i3, i4

Similarity weighted recommendation:

$$\frac{0.533 * 2 + 0.526 * 4 + 0.285 * 2}{0.533 + 0.526 + 0.285} = 2.78$$

Unweighted recommendation:

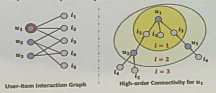
$$\frac{2 + 4 + 2}{3} = 2.67$$

Q4 Use mean for this tut

Q2

Q2

- High-order connectivity
  - Recommender systems rely on capturing similarity
    - User-user (User-CF), item-item (Item-CF), user-item (Model-CF)
  - GNN extends similarity to high-orders
    - Connectivity among high-order neighbors
  - Besides, data sparsity issue is well addressed



Answer

a) Equal frequency: #bins = 3, Frequency of each bin = 4

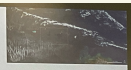
Bin#1:	Bin#2:	Bin#3
[5, 10, 11, 13]	[15, 35, 50, 55]	[72, 92, 204, 215]
Mean=10	Mean=39	Mean=146
[10, 10, 10, 10]	[39, 39, 39, 39]	[146, 146, 146, 146]

b) Equal Width = (215-5)/3 = 70

Bin#1: 5-75	Bin#2: 75-145	Bin#3: 145-215
[5, 10, 11, 13, 15, 35, 50, 55, 72]	[92]	[204, 215]
Mean = 30	Mean = 92	Mean = 210
[30, 30, 30, 30, 30, 30, 30, 30, 30]	[92]	[210, 210]

Q3

(1) Answers



Trying to detect oil slicks from satellite images to give early warning of ecological disasters and deter illegal dumping

This is essentially a classification task.

- Step 1: data cleaning, preprocessing
- Step 2: manually assign labels to images:
  1 for oil slick and **0** for " no oil slick"
- Step 3: Build a classifier, Train it using labeled data, test it over unlabeled data

Q5

Basic solution

- For each pair of entities (g, y), one from G and the other from Y
  - Compute the similarity of g and y
    - How?
  - Simple threshold method: If the distance below some number, same
  - Define some rule
- Complexity?
  - There are $O(N^2)$ possible matches
  - Each match take m
  - $O(m N^2)$

better efficiency search
better space
Use similar function

(2) Answers

A bank is trying to reduce customer attrition

- Step 1: Clean and preprocess data and select customers profile data, their transaction history
- Step 2: Perform classification on customers into "loyal" and "nonloyal"
- Step 3: Build a classifier, Train it using labeled data, test it over unlabeled data

(3) Answers

A supermarket is planning store layouts

It can be solved by associate rule mining.

- Step 1: Clean, preprocess and select transactional data
- Step 2: Perform Association Rule Mining (ARM)
- Step 3: Re-arrange the store based on ARM results in order to facilitate the customers

Example:
{diaper, milk} --> {beer}