COMP90042

Workshop Week 05

- Homework 2
  - Due: Sunday, April 1

- No workshops next week (Easter holiday break)

Syllabus

1	Introduction and Preprocessing	Text classification
2	Lexical semantics	Distributional semantics
3	Part of Speech Tagging	Hidden Markov Models
4	Unsupervised Hidden Markov Models	Context-Free Grammars
5	Probabilistic Parsing	Dependency parsing
	Easter holiday break	
6	N-gram language modelling	Deep learning for language models and tagging
7	Information Extraction	Question Answering
8	Topic Models	*ANZAC day holiday*
9	Information Retrieval -- Boolean search and the vector space model	Indexing and querying in the vector space model, evaluation
10	Index and vocabulary compression	Efficient query processing
11	The Web as a Graph: Page-rank & HITS	Machine Translation (word based)
12	Machine translation (phrase based) and neural encoder-decoder	Subject review

Outline

- Parsing
  - The Chomsky Normal Form (CNF)
  - The CYK algorithm

- Consider the following simple **context–free grammar**
  - S -> NP VP
  - VP -> V NP | V NP PP
  - PP -> P NP
  - V -> "saw" | "walked"
  - NP -> "John" | "Bob" | Det N | Det N PP
  - Det -> "a" | "an" | "the" | "my"
  - N -> "man" | "cat" | "telescope" | "park"
  - P -> "on" | "by" | "with“

- Parse the following sentence
  - "an park by Bob walked an park with Bob"

A possible parsing tree using the rules

<table>
<tr><td colspan="10">S</td></tr>
<tr><td colspan="4">NP</td><td colspan="6">VP</td></tr>
<tr><td rowspan="2">Det</td><td rowspan="2">N</td><td colspan="2">PP</td><td rowspan="2">V</td><td colspan="2">NP</td><td colspan="2">PP</td></tr>
<tr><td>P</td><td>NP</td><td>Det</td><td>N</td><td>P</td><td>NP</td></tr>
<tr><td>an</td><td>park</td><td>by</td><td>Bob</td><td>walked</td><td>an</td><td>park</td><td>with</td><td>Bob</td></tr>
</table>

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

The CYK parsing algorithm

- Convert grammar to Chomsky Normal Form (CNF)
- Fill in a parse table
- Use table to derive parse
- Convert result back to original grammar

Rewrite in Chomsky Normal Form (CNF)

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

In CNF, each rule consists of either:

a (single) non-terminal which re-writes as a single terminal, or

a (single) non-terminal which re-writes as exactly two non-terminals

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

Original rules

<table>
<tr><td colspan="9">S</td></tr>
<tr><td colspan="4">NP</td><td colspan="5">VP</td></tr>
<tr><td rowspan="2">Det</td><td rowspan="2">N</td><td colspan="2">PP</td><td rowspan="2">V</td><td colspan="2">NP</td><td colspan="2">PP</td></tr>
<tr><td>P</td><td>NP</td><td>Det</td><td>N</td><td>P</td><td>NP</td></tr>
<tr><td>an</td><td>park</td><td>by</td><td>Bob</td><td>walked</td><td>an</td><td>park</td><td>with</td><td>Bob</td></tr>
</table>

CNF (Binary tree)

<table>
<tr><td colspan="9">S</td></tr>
<tr><td colspan="4">NP</td><td colspan="5">VP</td></tr>
<tr><td rowspan="3">Det</td><td colspan="3">Y</td><td rowspan="3">V</td><td colspan="4">X</td></tr>
<tr><td rowspan="2">N</td><td colspan="2">PP</td><td colspan="2">NP</td><td colspan="2">PP</td></tr>
<tr><td>P</td><td>NP</td><td>Det</td><td>N</td><td>P</td><td>NP</td></tr>
<tr><td>an</td><td>park</td><td>by</td><td>Bob</td><td>walked</td><td>an</td><td>park</td><td>with</td><td>Bob</td></tr>
</table>

an	park	by	Bob	walked	an	park	with	Bob
[0,1]	[0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]	[0,9]
	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]	[1,9]
		[2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]	[2,9]
			[3,4]	[3,5]	[3,6]	[3,7]	[3,8]	[3,9]
				[4,5]	[4,6]	[4,7]	[4,8]	[4,9]
					[5,6]	[5,7]	[5,8]	[5,9]
						[6,7]	[6,8]	[6,9]
							[7,8]	[7,9]
								[8,9]

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

an	park	by	Bob	walked	an	park	with	Bob
[0,1]	[0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]	[0,9]
	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]	[1,9]
		[2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]	[2,9]
			[3,4]	[3,5]	[3,6]	[3,7]	[3,8]	[3,9]
				[4,5]	[4,6]	[4,7]	[4,8]	[4,9]
					[5,6]	[5,7]	[5,8]	[5,9]
						[6,7]	[6,8]	[6,9]
							[7,8]	[7,9]
								[8,9]

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

an	park	by	Bob	walked	an	park	with	Bob
Det [0,1]	[0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]	[0,9]
	N [1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]	[1,9]
		P [2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]	[2,9]
			NP [3,4]	[3,5]	[3,6]	[3,7]	[3,8]	[3,9]
				V [4,5]	[4,6]	[4,7]	[4,8]	[4,9]
					Det [5,6]	[5,7]	[5,8]	[5,9]
						N [6,7]	[6,8]	[6,9]
							P [7,8]	[7,9]
								NP [8,9]

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

an	park	by	Bob	walked	an	park	with	Bob
Det [0,1]	[0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]	[0,9]
	N [1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]	[1,9]
		P [2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]	[2,9]
			NP [3,4]	[3,5]	[3,6]	[3,7]	[3,8]	[3,9]
				V [4,5]	[4,6]	[4,7]	[4,8]	[4,9]
					Det [5,6]	[5,7]	[5,8]	[5,9]
						N [6,7]	[6,8]	[6,9]
							P [7,8]	[7,9]
								NP [8,9]

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

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

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

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

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

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

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

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

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

Figure 12.5 The CKY algorithm.

Two possible trees in CNF

<table>
<tr><td colspan="10">S</td></tr>
<tr><td colspan="4">NP</td><td colspan="6">VP</td></tr>
<tr><td rowspan="3">Det</td><td colspan="3">Y</td><td rowspan="3">V</td><td colspan="5">NP</td></tr>
<tr><td rowspan="2">N</td><td colspan="2">PP</td><td rowspan="2">Det</td><td colspan="4">Y</td></tr>
<tr><td>P</td><td>NP</td><td rowspan="2">N</td><td colspan="3">PP</td></tr>
<tr><td>P</td><td>NP</td></tr>
<tr><td>an</td><td>park</td><td>by</td><td>Bob</td><td>walked</td><td>an</td><td>park</td><td>with</td><td>Bob</td></tr>
</table>

<table>
<tr><td colspan="10">S</td></tr>
<tr><td colspan="4">NP</td><td colspan="6">VP</td></tr>
<tr><td rowspan="3">Det</td><td colspan="3">Y</td><td rowspan="3">V</td><td colspan="5">X</td></tr>
<tr><td rowspan="2">N</td><td colspan="2">PP</td><td colspan="3">NP</td><td colspan="2">PP</td></tr>
<tr><td>P</td><td>NP</td><td>Det</td><td>N</td><td>P</td><td>NP</td></tr>
<tr><td>an</td><td>park</td><td>by</td><td>Bob</td><td>walked</td><td>an</td><td>park</td><td>with</td><td>Bob</td></tr>
</table>

Convert result back to original grammar

S								
NP				VP				
Det	N	PP		V	NP			
		P	NP		Det	N	PP	
							P	NP
an	park	by	Bob	walked	an	park	with	Bob

S								
NP				VP				
Det	N	PP		V	NP		PP	
		P	NP		Det	N	P	NP
an	park	by	Bob	walked	an	park	with	Bob

The CYK parsing algorithm

- Convert grammar to Chomsky Normal Form (CNF)
- Fill in a parse table
- Use table to derive parse
- Convert result back to original grammar