Text Mining and Chatbots

Introduction

Thomas Gerald

January 9, 2026

Ressources (for exercices and lecture slides)

https://thomas-gerald.fr/TMC/index.html

Text Mining And Chatbots: Schedule

- 09/01/2026 - Introduction (Thomas Gerald)
- 15/01/2026 - Tokenization (Thomas Gerald)
- 23/01/2026 - Semantics and Word Embeddings (Sahar Ghannay)
- 30/01/2026 - Open Domain vs. Specialty Domain (Sahar Ghannay)
- 06/02/2026 - Langage models and the transformer (Thomas Gerald)
  → language model, transformer architecture, training, fine tuning and adaptation
- 13/02/2026 - State-of-the-Art and applications (Thomas Gerald)
  → Linear based attention, model optimisation, other approach for language model
- 20/02/2026 - Defense

Text Mining And Chatbots: Sessions organisation

Typical Session

1. A lecture ($\approx$ 1 to 2 hours)
2. A lab exercices
3. A report to submit before the next session

Grading

- Exercise reports (50%)
- Defense (50%)

Reports

Submit reports based on the exercise lab

- You will be in group (2 to 3 person per group)
- Modality of the submission will depend on the teacher

The Defense

Oral presentation of an article from ACL

- You will be in group (3 to 4 person per group)
- 10 minutes presentation, 5 minutes questions
- We will forward you a list of articles
- You can propose an article to defend but you need our validation

Examples of presented articles (previous year):

- "Rethinking Tabular Data Understanding with Large Language Models", Tianyang Liu, Fei Wang, and Muhao Chen. NAACL 2024
- "From Good to Great: Improving Math Reasoning with Tool-Augmented Interleaf Prompting", Nuo Chen, Hongguang Li, Baoyuan Wang, and Jia Li., NLRSE 2024
- "QLORA:Efficient Finetuning of Quantized LLMs", Tim Dettmers, Artidoro Pagnoni, Ari Holtzman and Luke Zettlemoyer, NeurIPS 2023
- "The Curious Case of Neural Text Degeneration", Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes and Yejin Choi, ICLR 2020
- "Predicting Entity Salience in Extremely Short Documents,Benjamin Bullough, Harrison Lundberg, Chen Hu, and Weihang Xiao", EMNLP 2024

NLP Introduction

NLP tasks

Pre-processing tasks:

Prepare documents

- Remove non-text or irrelevant content
- tokenization, cleaning, stopword removal, stemming, lemmatization (*Next session*)

Low level tasks:

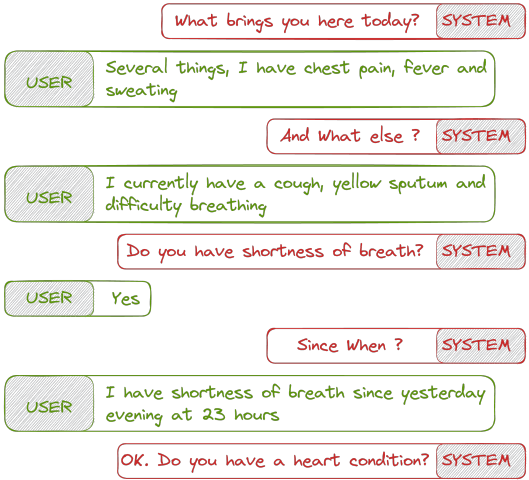
based on words, keywords $\rightarrow$ serve futur tasks

- Document classification
- Sentence/chunk identification
- Part-of-speech (`POS`) tagging
- Named entity recognition (NER), relationship extraction (e.g, anaphora resolution), word sense disambiguation...

NLP tasks

High level tasks:

- Information Retrieval (IR)
- Question Answering (QA)
- Automatic Translation (MT)
- Natural Language Understanding (NLU)
- Sentiment Analysis
- Dialogue systems



Short history of approaches

History: Automatic Translation (MT)

- **1954 :** A first system for Russianto English Translation (cold war historical context)
  - 250 “words”
  - 6 grammars rules
  - 49 selected samples
- **1954 (march):** Scientific journal “Mechanical Translation” founded
- **1954-1966:**
  - new methods based on interlingual representation
  - Stating new problems → context dependent, commensense reasoning:

    “The box was is in the **pen**”

    → **“pen”** def 1 : writting tool

    → **“pen”** def 2 : corral, coop, animal enclosure

    Conclusion (Bar-Hillel): No computer programs could conceivably deal with such “real-world” knowledge without recourse of to large encyclopedic ressources.

History: Automatic Translation (MT)

- **1966 :** The ALPAC (Automatic Language Processing Advisory Commitee) report
  $\rightarrow$ There is no immediat or predictable prospect of usefull machine translation
  $\rightarrow$ Stop funding MT (at least in the US)
- **1967-2000**
  - Specific Domain Translation e.g TAUM Project
  - First commercial systems (Systran, ALPS) - toolbox for translation
  - Multilingual transfer system (phrase structure rule, dependency grammar, statistical approaches)
- **2000**: Statistical Machine Translation (raise of internet)
- **2010-Today** Neural approaches
  - RNN for Translation
  - LLMs approaches

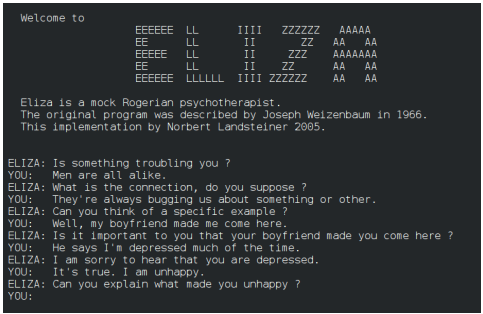
History: Language Structure

- **1954:** (Zellig Haris) Distribution structure of languages: consistency + distributional relations between words and sentences
- **1957:** Noam Chomsky, syntactic structure, generative grammar/formal languages
  → "Colorless green ideas sleep furiously"

is the syntax ok? is the meaning ok?

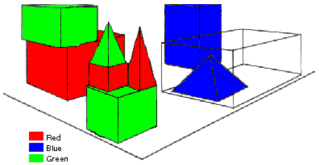
History: Language Structure

- **1956:** McCarthy, Minsky, Newell, Simon: computer with language abilities
  - **1961** – BASEBALL
  - **1964** – SIR
  - **1964** – STUDENT
  - **1966** – ELIZA



History: Language Structure

- Knowlege representation: Quillian (semantic networks)
  → 1968-70: SCHRDLU (Terry Winograd MIT); first system that "understands"



Person: Pick up a big red block.

Computer: OK.

Person: Grasp the pyramid.

Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.

Person: Find a block which is taller than the one you are holding and put it into the box.

Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.

Computer: OK.

https://science.slc.edu/ jmarshall/courses/cs30/Lectures/week10/SHRDLU.html

NLP: A small reminder

Neural approaches are not recents !!!

- **1957:** The perceptron (neural networks)
- **1980:** The neocognitron (Ancestor of Convolutional neural networks)
- **1986:** Gradient descent (backpropagation algorithm) for neural networks

$\rightarrow$ **But few computational power**

NLP tasks

NLP tasks: A first example

The president of the United States was eating an apple with a knife.

How to process? What pipeline?

NLP tasks: A first example

The president of the United States was eating an apple with a knife.

How to process? What pipeline?

- Text segmentation/tokenization $\rightarrow$ lexical units

NLP tasks: A first example

The president of the United States was eating an apple with a knife.

How to process? What pipeline?

- Text segmentation/tokenization → lexical units
- Recognition of the lexical components and their properties → lexical processing (lemmatization...);

NLP tasks: A first example

The president of the United States was eating an apple with a knife.

How to process? What pipeline?

- Text segmentation/tokenization → lexical units
- Recognition of the lexical components and their properties → lexical processing (lemmatization...);
- recognition of the higher level components, and the relations between them → syntactic processing;

NLP tasks: A first example

The president of the United States was eating an apple with a knife.

How to process? What pipeline?

- Text segmentation/tokenization $\rightarrow$ lexical units
- Recognition of the lexical components and their properties $\rightarrow$ lexical processing (lemmatization...);
- recognition of the higher level components, and the relations between them $\rightarrow$ syntactic processing;
- building the meaning representation of the statement $\rightarrow$ semantic analysis;

NLP tasks: A first example

The president of the United States was eating an apple with a knife.

How to process? What pipeline?

- Text segmentation/tokenization $\rightarrow$ lexical units
- Recognition of the lexical components and their properties $\rightarrow$ lexical processing (lemmatization...);
- recognition of the higher level components, and the relations between them $\rightarrow$ syntactic processing;
- building the meaning representation of the statement $\rightarrow$ semantic analysis;
- how this statement may be related to the context in which it is analyzed ? (text, dialog,...) $\rightarrow$ pragmatics.

Document splitting: Split into sentences?

M. Martin will defend his PhD entitled "Integration of NLP stuff in unsteady AI models for low-level tasks" on January 13th, 2023, at 2.30pm The defense will take place in room A.009, building 507. Link for live broadcast available on demand

Document splitting: Split into sentences?

M. Martin will defend his PhD entitled "Integration of NLP stuff in unsteady AI models for low-level tasks" on January 13th, 2023, at 2.30pm The defense will take place in room A.009, building 507. Link for live broadcast available on demand

→ M. Martin will defend his PhD entitled "Integration of NLP stuff in unsteady AI models for low-level tasks" on January 13th, 2023, at 2.30pm

Document splitting: Split into sentences?

M. Martin will defend his PhD entitled "Integration of NLP stuff in unsteady AI models for low-level tasks" on January 13th, 2023, at 2.30pm The defense will take place in room A.009, building 507. Link for live broadcast available on demand

→ M. Martin will defend his PhD entitled "Integration of NLP stuff in unsteady AI models for low-level tasks" on January 13th, 2023, at 2.30pm
→ The defense will take place in room A.009, building 507.

Document splitting: Split into sentences?

M. Martin will defend his PhD entitled "Integration of NLP stuff in unsteady AI models for low-level tasks" on January 13th, 2023, at 2.30pm The defense will take place in room A.009, building 507. Link for live broadcast available on demand

→ M. Martin will defend his PhD entitled "Integration of NLP stuff in unsteady AI models for low-level tasks" on January 13th, 2023, at 2.30pm
→ The defense will take place in room A.009, building 507.
→ Link for live broadcast available on demand

NLP tasks : Tokenization

Transform text into lexical units (tokens)

- Splitting around space
- Adding space arround punctuation mark

2024-01-09: he developed dyspnea on exertion when climbing stairs, asso w/ 2-pillow orthopnea and PND.

NLP tasks : Tokenization

Transform text into lexical units (tokens)

- Splitting around space
- Adding space arround punctuation mark

2024-01-09: he developed dyspnea on exertion when climbing stairs, asso w/ 2-pillow orthopnea and PND.

↓

Transform text into lexical units (tokens)

- Splitting around space
- Adding space arround punctuation mark

2024-01-09: he developed dyspnea on exertion when climbing stairs, asso w/ 2-pillow orthopnea and PND.

↓

2024-01-09 : he developed dyspnea on exertion when climbing stairs , assoc w/ 2 - pillow orthopnea and PND .

When language does not have space...

Splitting a sequence of Chinese characters into meaningful words

电梯小

When language does not have space...

Splitting a sequence of Chinese characters into meaningful words

电梯小

↓

When language does not have space...

Splitting a sequence of Chinese characters into meaningful words

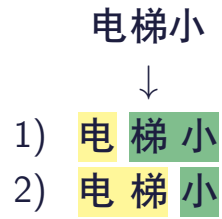
电梯小

↓

1) 电 梯 小

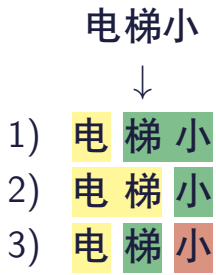
When language does not have space...

Splitting a sequence of Chinese characters into meaningful words



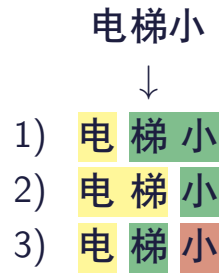
When language does not have space...

Splitting a sequence of Chinese characters into meaningful words



When language does not have space...

Splitting a sequence of Chinese characters into meaningful words



- 电 梯 ← elevator
- 小 ← small

NLP tasks : Tokenization

Tokenization:

1. Before ≈ 2016, classical methods consisted in surrounding punctuation with space to produce tokens
2. Deep-learning approaches today consider statistical approach to splits words into subwords (not always have a meaning)
   - constitution $\rightarrow$ _cons + titu + tion
   - Different statistical algorithms relying on frequencies of characters string in a corpus:
     - **BPE**: Byte Pair Encoding
     - **SentencePiece**

   $\rightarrow$ Next session !

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates $\rightarrow$ unfortunate

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates $\rightarrow$ unfortunate
  - malheureuses $\rightarrow$ malheureux

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates $\rightarrow$ unfortunate
  - malheureuses $\rightarrow$ malheureux
- nouns: singular form

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates → unfortunate
  - malheureuses → malheureux
- nouns: singular form
  - eyes → eye / yeux → œil

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates → unfortunate
  - malheureuses → malheureux
- nouns: singular form
  - eyes → eye / yeux → œil
  - mice → mouse / souris → souris

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates → unfortunate
  - malheureuses → malheureux
- nouns: singular form
  - eyes → eye / yeux → œil
  - mice → mouse / souris → souris
- verbs: infinitive form

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates → unfortunate
  - malheureuses → malheureux
- nouns: singular form
  - eyes → eye / yeux → œil
  - mice → mouse / souris → souris
- verbs: infinitive form
  - was → be / fus → être

NLP tasks : lemmatization

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates → unfortunate
  - malheureuses → malheureux
- nouns: singular form
  - eyes → eye / yeux → œil
  - mice → mouse / souris → souris
- verbs: infinitive form
  - was → be / fus → être

→ Ambiguities from surface form to lemmatisation

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates → unfortunate
  - malheureuses → malheureux
- nouns: singular form
  - eyes → eye / yeux → œil
  - mice → mouse / souris → souris
- verbs: infinitive form
  - was → be / fus → être

→ Ambiguities from surface form to lemmatisation

- Les souris dansent → le[DT] souris[NN] danser[VB]

Lemmatization (lemmatisation):

Transform words/tokens into canonical form (or lemma)

- adjectives: singular form (or select a gender for some language)
  - unfortunates → unfortunate
  - malheureuses → malheureux
- nouns: singular form
  - eyes → eye / yeux → œil
  - mice → mouse / souris → souris
- verbs: infinitive form
  - was → be / fus → être

→ Ambiguities from surface form to lemmatisation

- Les souris dansent → le[DT] souris[NN] danser[VB]
- Tu souris souvent → tu[PRP] sourire[VB] souvent[ADV]

NLP tasks : Stemming

Stemming (Racinisation):

A stem is a morphological subpart of a word that exclude flexional endings

- studying $\rightarrow$ stud (VB)
- students $\rightarrow$ stud (NN)
- chantions $\rightarrow$ chant (VB)
- chanteuses $\rightarrow$ chant (NN)

Often automatic process:

Using the Porter Stemmer:

"he was eating an apple" $\rightarrow$ he wa eat an appl

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)
- **Pro** Pronoun (she, we,...)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)
- **Pro** Pronoun (she, we,...)
- **Det** Determinent (the, a)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)
- **Pro** Pronoun (she, we,...)
- **Det** Determinent (the, a)
- **Verb** verb (eat, read, saw)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)
- **Pro** Pronoun (she, we,...)
- **Det** Determinent (the, a)
- **Verb** verb (eat, read, saw)
- **Punct** punctuation (., :, ...)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)
- **Pro** Pronoun (she, we,...)
- **Det** Determinent (the, a)
- **Verb** verb (eat, read, saw)
- **Punct** punctuation (., :, ...)
- **Num** Number (14, 200,...)

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)
- **Pro** Pronoun (she, we,...)
- **Det** Determinent (the, a)
- **Verb** verb (eat, read, saw)
- **Punct** punctuation (., :, ...)
- **Num** Number (14, 200,...)

Studying	NLP	is	Nice	!	They	study	NLP	this	year	.

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (basic):

Morphological label for each tokens :

- **Adj** adjectives (small, strong,...)
- **Adv** adverbs (truly, seriously,...)
- **Conj** conjunctions (if, and, until,...)
- **Noun** nouns (an apple, a saw,...)
- **Pro** Pronoun (she, we,...)
- **Det** Determinent (the, a)
- **Verb** verb (eat, read, saw)
- **Punct** punctuation (., :, ...)
- **Num** Number (14, 200,...)

Studying	NLP	is	Nice	!	They	study	NLP	this	year	.
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Verb	Noun	Verb	Adj	Punct	Pro	Verb	Noun	Det	Noun	Punct

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (detailed):

Morphological label for each tokens :

- **VB** Verb, base form (to eat,...)
- **VBD** Verb, past tense (ate, rated)
- **VBG** Verb, gerund or present participle (eating, rating)
- **NN** Noun, singular or mass
- **NNP** Proper noun, singular
- **PRP** Personal pronoun (he, they)

⋮

NLP tasks : PoS (Part Of Speech Tagging)

PoS tagging (detailed):

Morphological label for each tokens :

- **VB** Verb, base form (to eat,...)
- **VBD** Verb, past tense (ate, rated)
- **VBG** Verb, gerund or present participle (eating, rating)
- **NN** Noun, singular or mass
- **NNP** Proper noun, singular
- **PRP** Personal pronoun (he, they)

⋮

Studying	NLP	is	Nice	!	They	study	NLP	this	year	.

PoS tagging (detailed):

Morphological label for each tokens :

- **VB** Verb, base form (to eat,...)
- **VBD** Verb, past tense (ate, rated)
- **VBG** Verb, gerund or present participle (eating, rating)
- **NN** Noun, singular or mass
- **NNP** Proper noun, singular
- **PRP** Personal pronoun (he, they)

⋮

Studying	NLP	is	Nice	!	They	study	NLP	this	year	.
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
VBG	NN	VBZ	Adj	Punct	PRP	Verb	NN	Det	NN	Punct

NLP tasks : PoS (Part Of Speech Tagging)

`https://universaldependencies.org/u/pos/` → universal POS tags

- **Open class words (non-finite set)**: ADJ, ADV, INTJ, NOUN, PROPN, VERB
  → **I postagged wikipedia** (creation of a new verb!!!)
- **Closed class words: (finite set)**: ADP, AUX, CCONJ (coordinating conjuction - and, yet,...), DET, NUM, PRON, SCONJ (subordinating conjunction - because, after,...)

Annotated Corpora :

- **ENGLISH:** EWT (English Web Treebank), GUM (Georgetown University Multilayer corpus)
- **FRENCH:** Sequoia, ParisStories (oral transcriptions of monologues and dialogues of students living in the Parisian region), Rhapsodie (spoken French)

NLP tasks : PoS (Part Of Speech Tagging)

Different label set to cover language specificities:

GRACE action to evaluate French morpho-syntactic taggers, using detailed labels, see `https://thomaslebarbe.wordpress.com/wp-content/uploads/2016/12/adda-et-al.pdf`

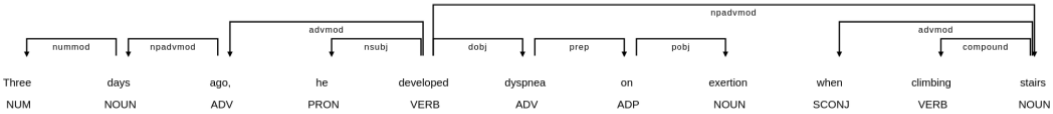
- Ncms = common noun (substantive), masculine, singular → soleil (sun)
- Npfs- = proper name (name), feminine, singular → Église (Church) = institution
- Vapp– = auxiliary verb, present participle → ayant (having), étant (being)
- Afpms = qualificative adjective, masculine, singular → grand
- Da-ms-d = definite article (determiner), masculine, singular → le (the)

More details:
`https://perso.limsi.fr/pap/inalco/TNML3_2020_2021/course_S1_9.pdf`

NLP task: Syntactic parsing

Dependency parsing:

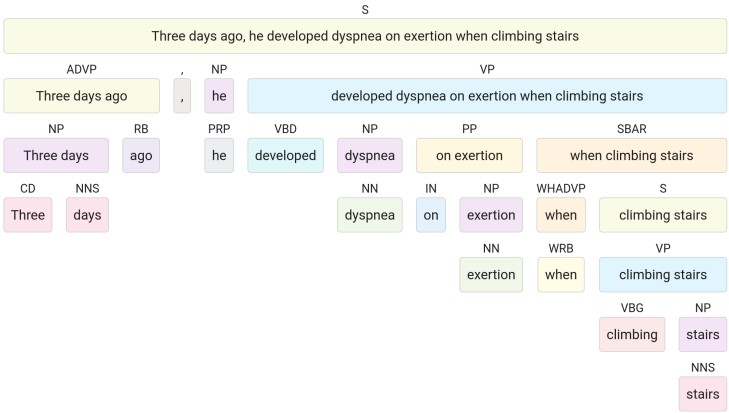
Identifying the grammatical relationships between words in a sentence, such as subject-verb relationships



NLP task: Syntactic parsing

Constituency parsing:

Identifying the constituent structure of a sentence, such as noun phrases and verb phrases.

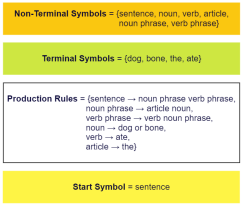


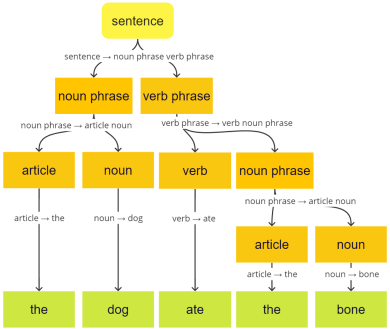
https://parser.kitaev.io/

Constituency parsing, how ?

- Sequential models
- hierachical models
- ...

→ based on a grammar (context-free grammar)





Short Introduction to grammar and Syntactic Parsing

Syntactic parsing

- Traditionnal syntactic parsing
- Historically (in NLP) based on Chomsky's generative paradigm
  - *The purpose of grammar is to generate all the possible statements in a language, and only them (grammatical statements*
  - Parsing a sentence means: associate to a grammatical statement(a sentence) its syntactic structure, i.e. the sequences of rewriting which has been necessary to obtain this statement from the initial axiom S of the grammar.

Generative grammar

Generative grammar: definiton

- G = (V_n, V_t, R,S), with :
  - V_n: non terminal vocabulary (what is needed to describe the language)
  - V_t: terminal vocabulary (the language words)
  - R: a set of rewriting rules, $X \rightarrow Y$
  - S : axiom of the grammar
- and depending of the rules in R :
  - unconstrained grammar: $\rightarrow$ too "loose"
  - contextual grammar : "X is rewrittent Y in the context u v" or $uXv \rightarrow uYv$:
  - context-free grammar : $X \rightarrow Y$, with X : 1 unique symbol from V_n
  - regular grammar: too strict (programming language)

Context-free Grammar et syntactic tree: Example

le garçon va lire un livre/ the boy will read a book

- $S \rightarrow NP\ VP$

Mais aussi : le pomme va mange la livre...

Context-free Grammar et syntactic tree: Example

le garçon va lire un livre/ the boy will read a book

- $S \rightarrow NP\ VP$
- $NP \rightarrow Det\ N$

Mais aussi : le pomme va mange la livre...

Context-free Grammar et syntactic tree: Example

le garçon va lire un livre/ the boy will read a book

- $S \rightarrow NP\ VP$
- $NP \rightarrow Det\ N$
- $VP \rightarrow (Aux)\ V\ NP$

Mais aussi : le pomme va mange la livre...

Context-free Grammar et syntactic tree: Example

le garçon va lire un livre/ the boy will read a book

- $S \rightarrow NP\ VP$
- $NP \rightarrow Det\ N$
- $VP \rightarrow (Aux)\ V\ NP$

Mais aussi : le pomme va mange la livre...

Context-free Grammar et syntactic tree: Example

le garçon va lire un livre/ the boy will read a book

- $S \rightarrow NP\ VP$
- $NP \rightarrow Det\ N$
- $VP \rightarrow (Aux)\ V\ NP$
- $Aux \rightarrow$ va
- $V \rightarrow$ lire|bat|mange| ...
- $Det \rightarrow$ le|la|les|un|...
- $N \rightarrow$ garçon|livre|pomme|...

Mais aussi : le pomme va mange la livre...

Context-free Grammar et syntactic tree: Example

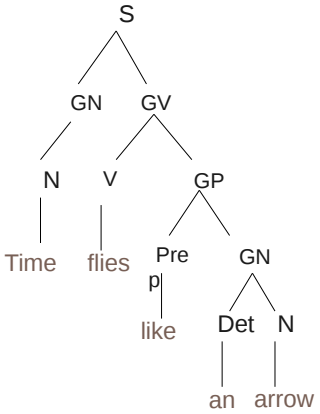
le garçon va lire un livre/ the boy will read a book

- $S \rightarrow NP\ VP$
- $NP \rightarrow Det\ N$
- $VP \rightarrow (Aux)\ V\ NP$
- $Aux \rightarrow$ va
- $V \rightarrow$ lire|bat|mange| ...
- $Det \rightarrow$ le|la|les|un|...
- $N \rightarrow$ garçon|livre|pomme|...

Mais aussi : le pomme va mange la livre...

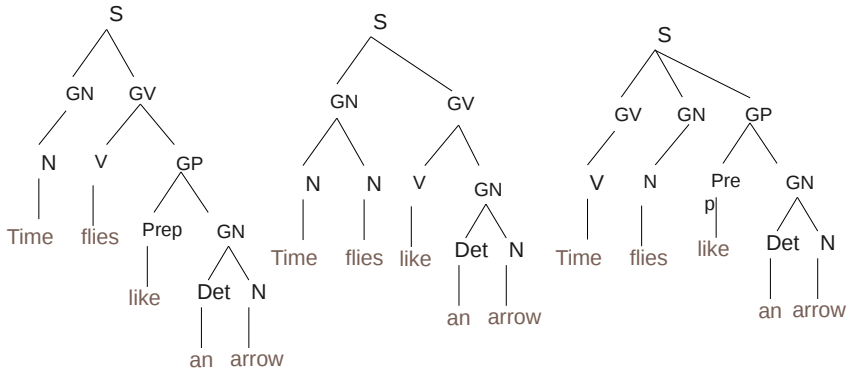
Context-free Grammar and syntactic tree: Example

- Differences between surface structure and deep structures
- Example from Chomsky: Time flies like an arrow :



Context-free Grammar and syntactic tree: Example

- Differences between surface structure and deep structures
- Example from Chomsky: Time flies like an arrow :



Formal Language Theory from Chomsky

- More a mathematical formalisation than a linguistic theory
- Language is not context free
  - grammatical agreements
- Contextual Grammars are not sufficient
  - Discontinuous constituents: *Combien cette salle a-t'elle de fenêtres ?*, the book that Mary wants you to read

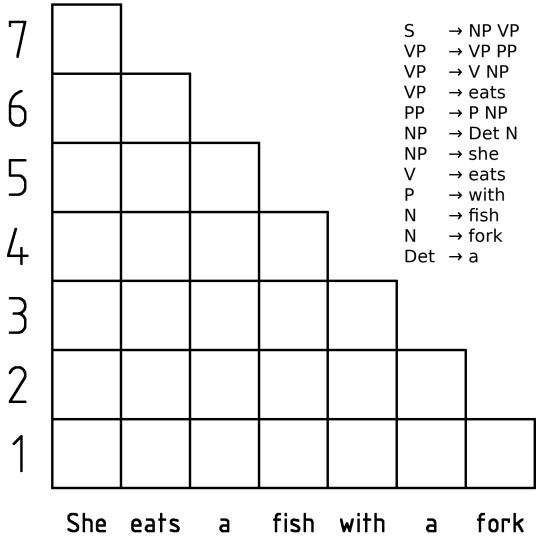
An algorithm to recognize contextual grammar

Cocke–Younger–Kasami Algorithm:

A parsing algorithm for context-free grammar (1961). A recursive algorithm.

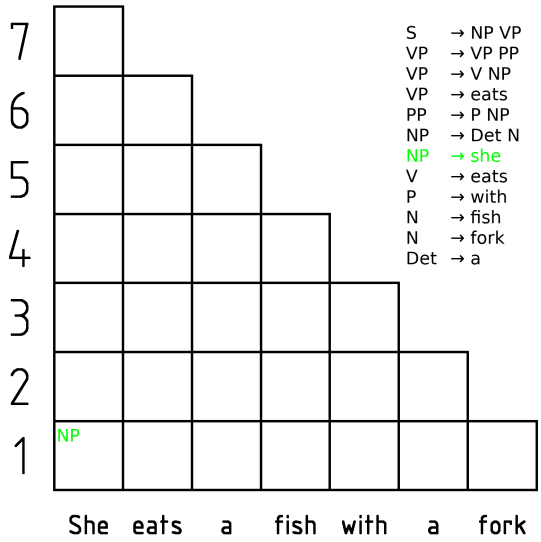
1. Find production rules that cover a token in s
2. Find rules that cover two tokens in s using 1.
3. ...
4. Find rules that cover n tokens in s using N-1 step.

CYK Algorithm

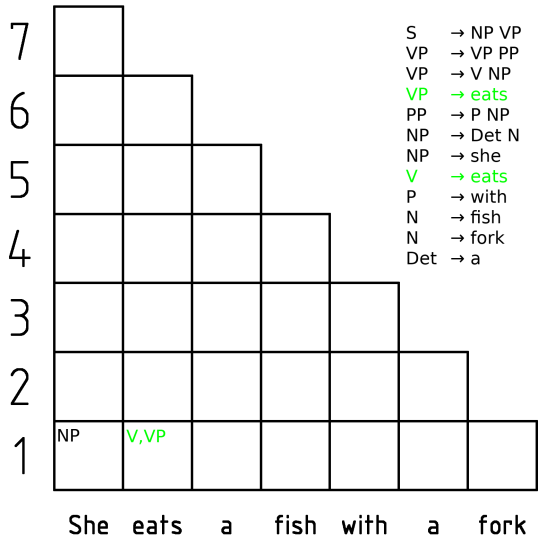


S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

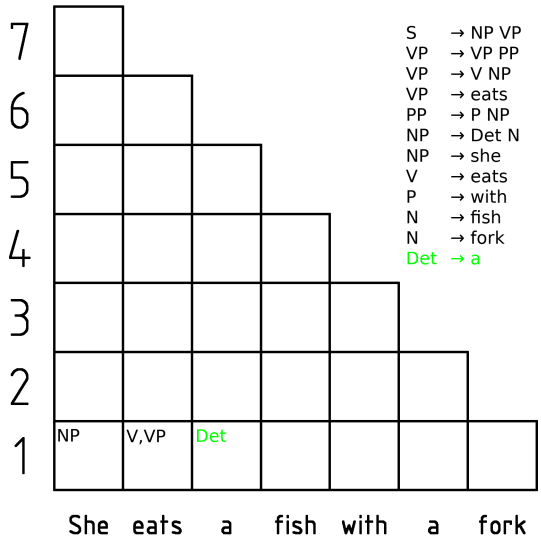


CYK algorithm



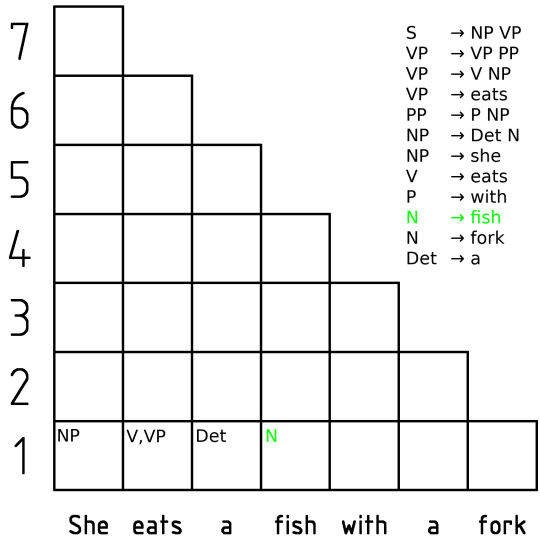
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm



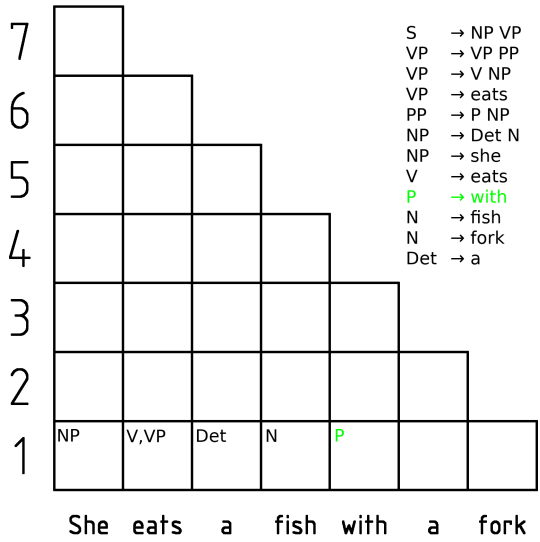
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm



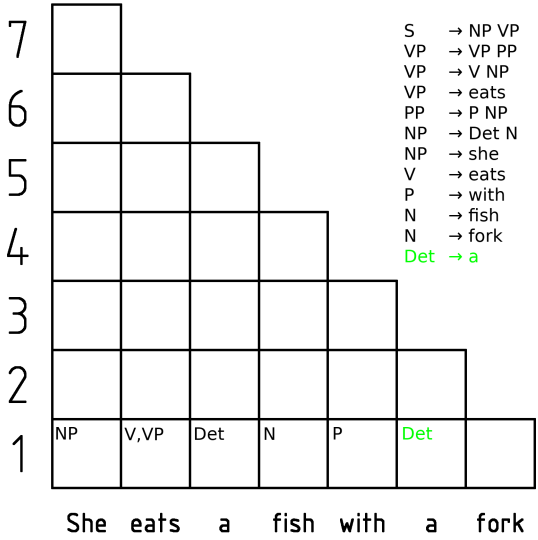
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm



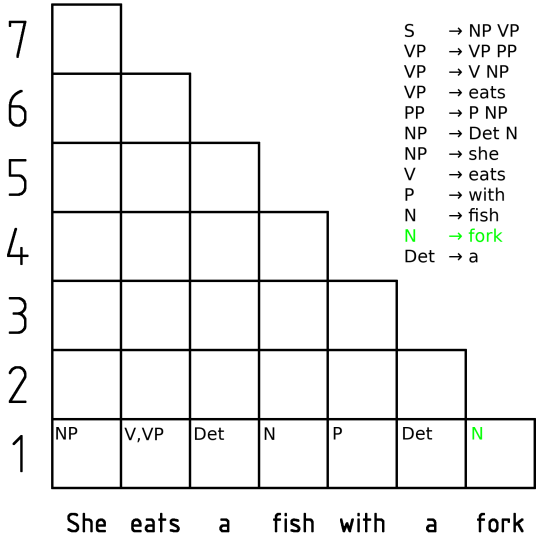
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm



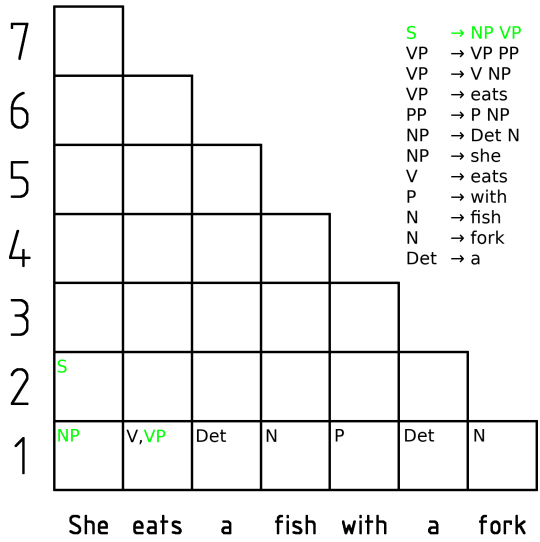
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm



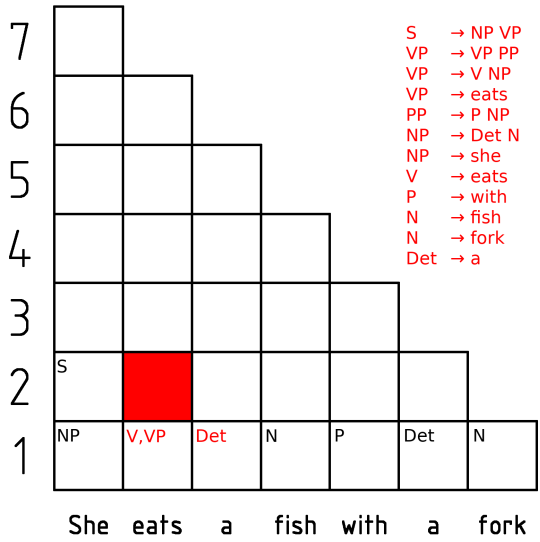
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm



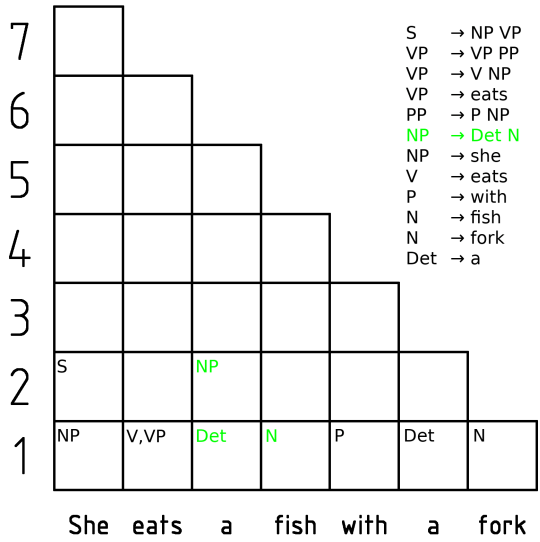
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm



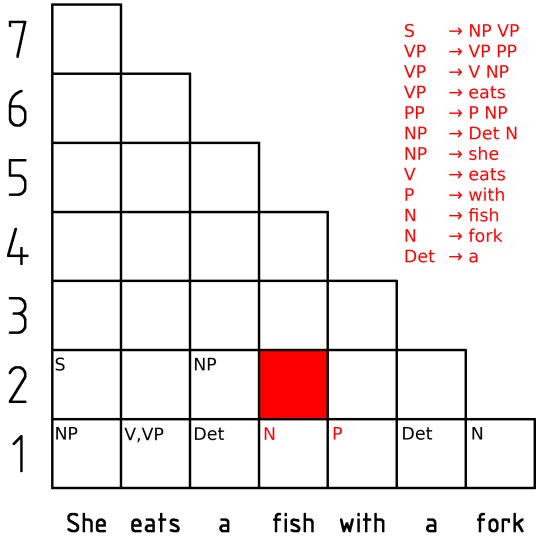
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

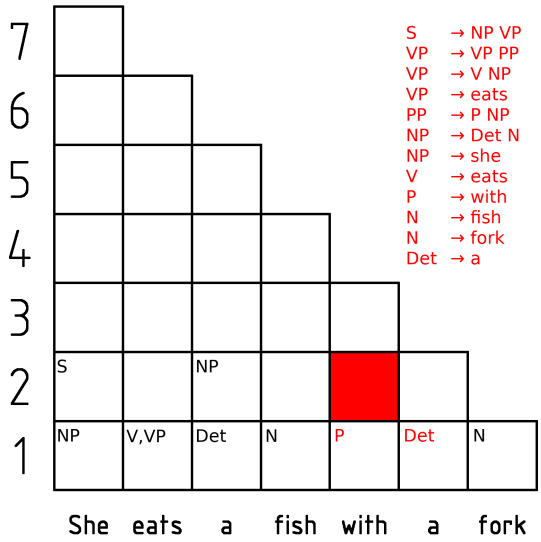


S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

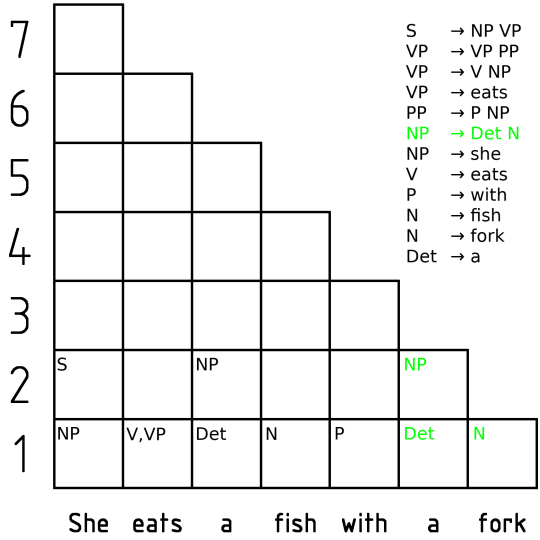
CYK algorithm



CYK algorithm

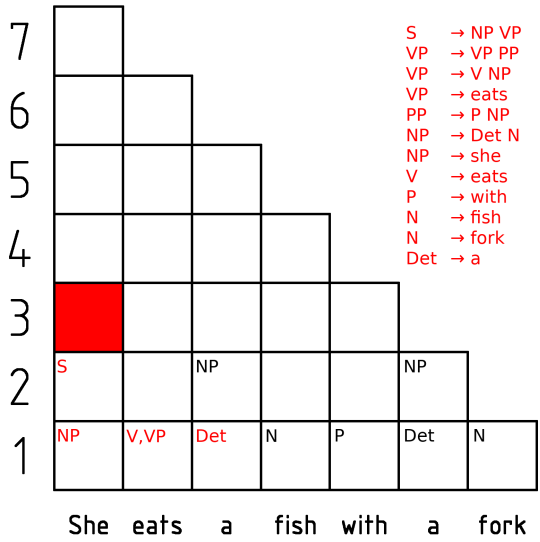


CYK algorithm



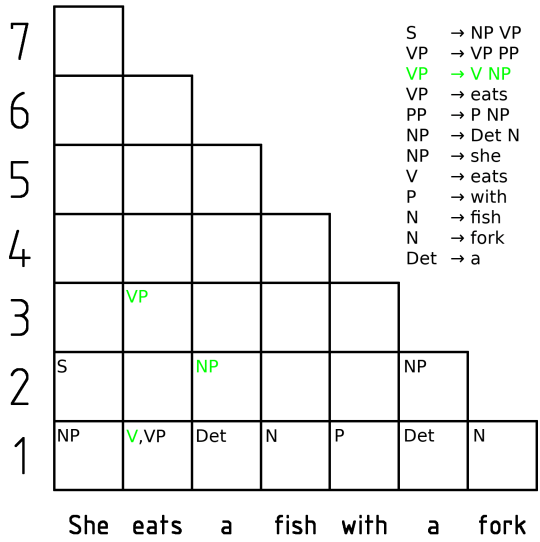
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

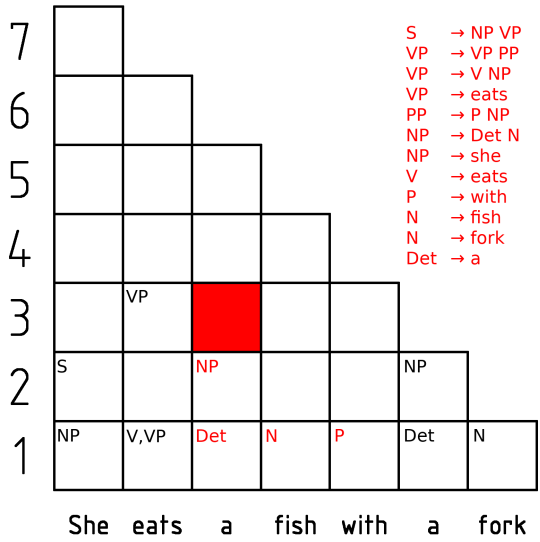


S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

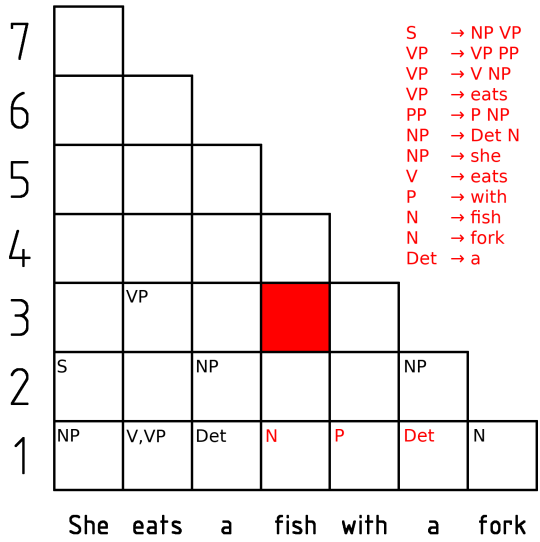


CYK algorithm

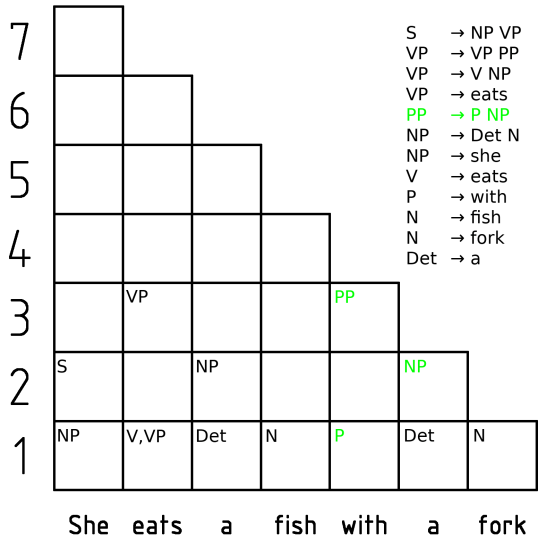


S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

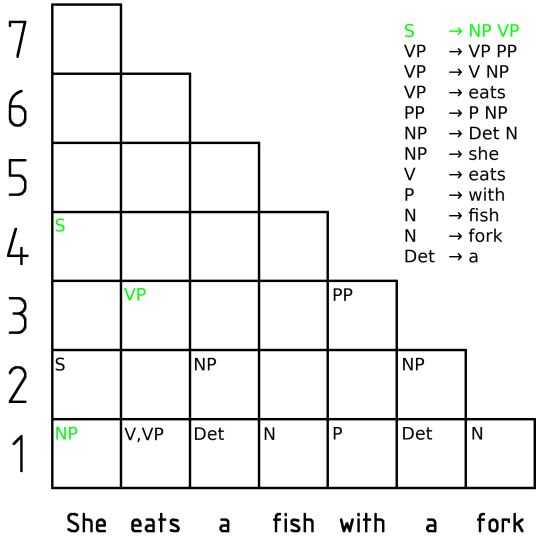
CYK algorithm



CYK algorithm

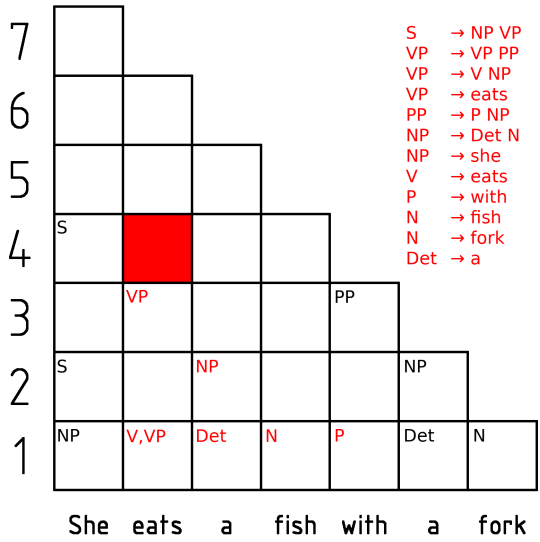


CYK algorithm

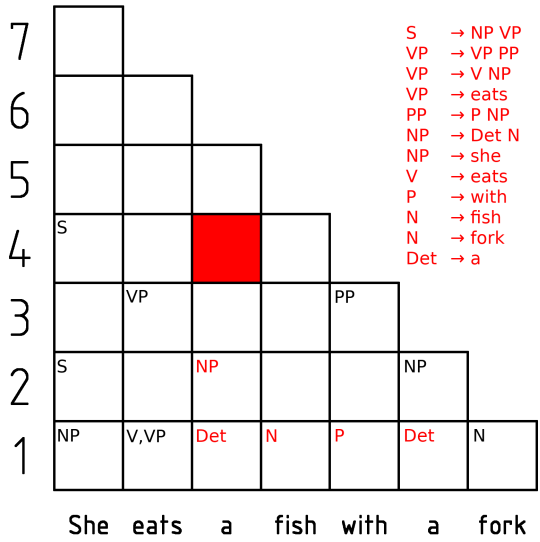


S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

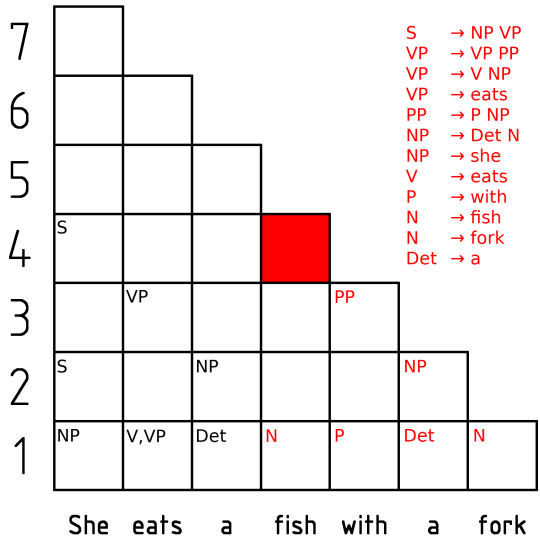


CYK algorithm



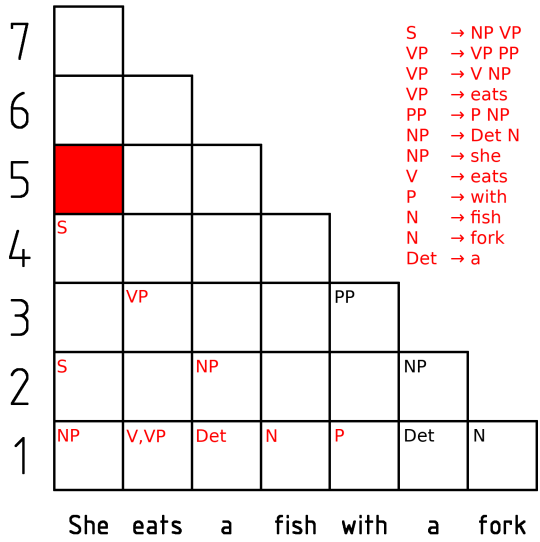
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

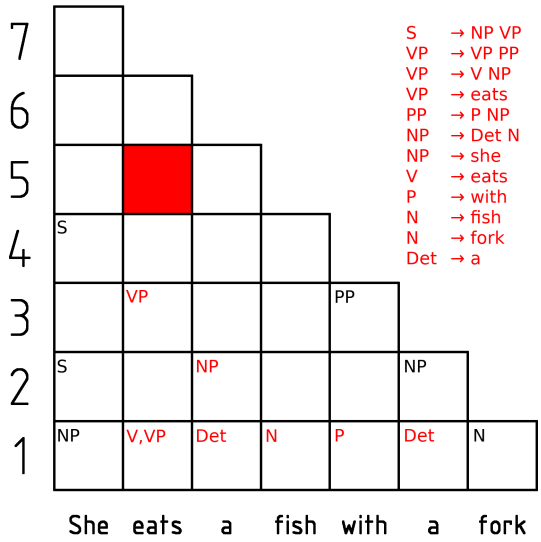


S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

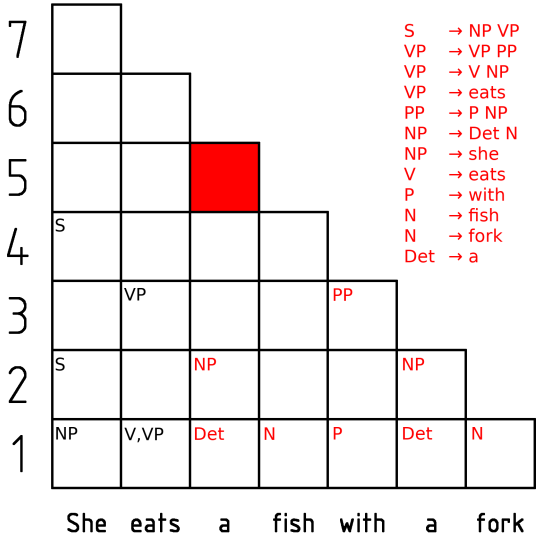


CYK algorithm



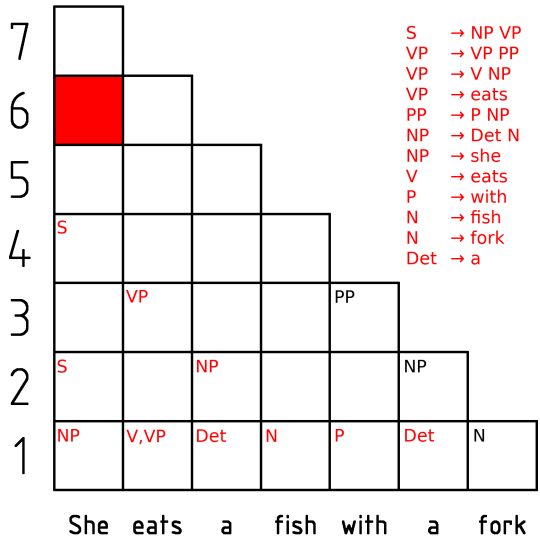
S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

CYK algorithm

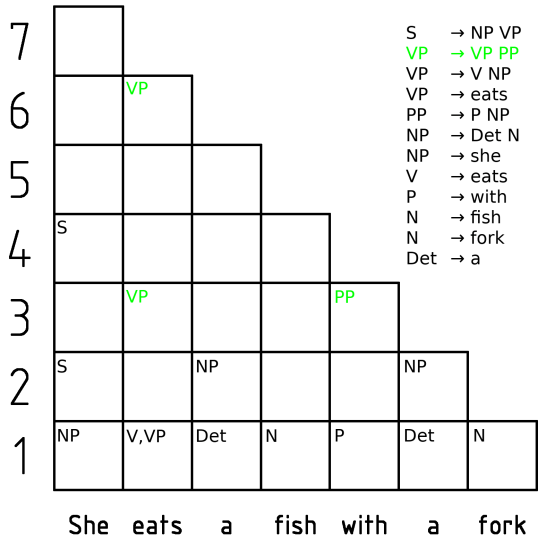


S → NP VP
VP → VP PP
VP → V NP
VP → eats
PP → P NP
NP → Det N
NP → she
V → eats
P → with
N → fish
N → fork
Det → a

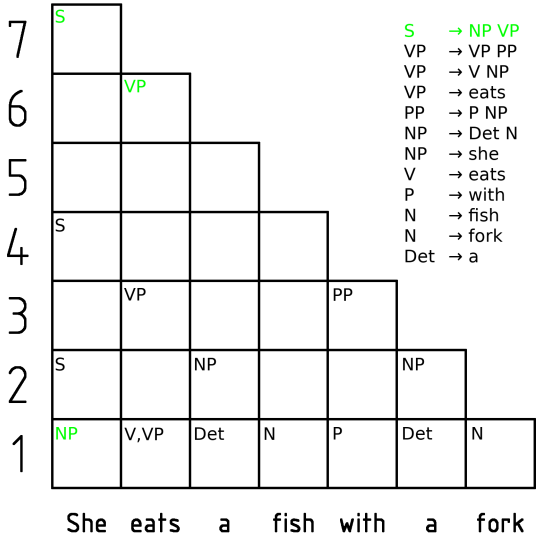
CYK algorithm



CYK algorithm



CYK algorithm



Exercise

Instruction

- Work together in small groups (up to 3 students per group)
- Prepare a final report composed of a header and three sections:
  - Header: first and last names and e-mail of all members of the group
  - First part: manual analysis of sentences
  - Second part: automatic analysis
  - Third part: observations
- Send your report (PDF) to thomas.gerald@universite-paris-saclay.fr (up to January 17th)

Instruction

- Gold standard annotation (reference, expected output):
  - Part-of-speech: ADJ, ADP, AUX, DET, NOUN, NUM, PROPN, PUNCT, SCONJ, VERB
  - Syntactic phrases: noun-phrases, verb-phrases
- Application of two POS tagging models on the dataset
  - spaCy (https://spacy.io/): pip install -U spacy
  - sciSpaCy (https://allenai.github.io/scispacy/): pip install scispacy
- Evaluation of produced outputs and discussion of results (don't forget to specify which model you have used among all available versions)

Spacy

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Time flies like an arrow.")
for token in doc:
    print(token.text, token.pos_, token.dep_)
```

Dataset

- Sentences from general domain with linguistics ambiguities:
  - The cat sat on teh couch.
  - Time flies like an arrow.
  - The spy saw the cop with the telescope.
  - The spy saw the cop with the revolver.
- Sentences from specialty domains:
  - Biology: Arabidopsis thaliana seedlings exhibit longer hypocotyls when they are grown under high ambient temperature, which is defined as thermomorphogenesis.
  - Astronomy: A spectrogram of PSN J10354824+3900279 obtained on Dec. 19.33 UT suggests that this is a type-Ia at redshift z 0.044.