



SYNTAX

Luísa Coheur

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- Syntax
 - Grasp fundamental concepts and learn how to perform a Syntactic Analysis

TOPICS

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

MAIN CONCEPTS

- Natural Language syntax restricts the sequences of words that are part of the language, but is much more flexible than the syntax of artificial languages
- Some used tags:
 - Noun Phrases (NP)
 - Verb Phrases (VP)
 - Prepositional Phrases (PP)
 - ...

MAIN CONCEPTS

- The used tags can be more functional:
 - Subject:
 - Example:
 - [The student]_{SUBJ} took the test.
 - Direct Object/Complement:
 - Example:
 - The student is reading [the book]_{DO}.
 - Indirect Object/Complement:
 - Example:
 - Give [the book]_{DO} [to Mary]_{IO}.
 - Predicative of the Subject:
 - Examples:
 - The teacher is [tired]_{PS}.
 - Maria is [a teacher]_{PS}.

EXAMPLE

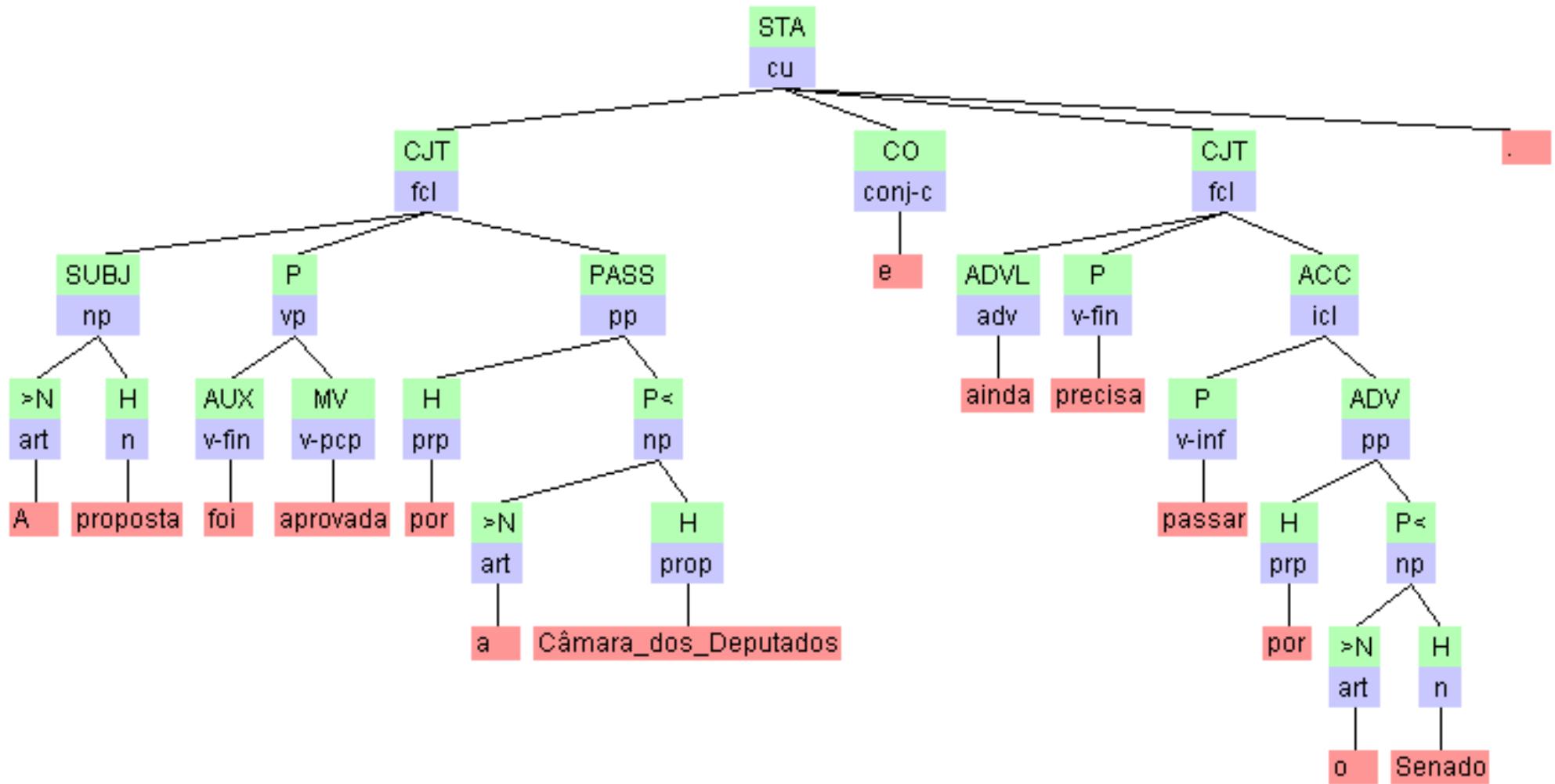
Wow, Sarah carefully hands her friend the red book from the shelf, and smiles.

- Interjection (INTJ):
- Subject (NP):
- Verb Phrase (VP):
 - Adverb (ADV):
 - Verb (V):
 - Indirect Object (NP):
 - Direct Object (NP):
 - ...
 - Prepositional Phrase (PP):
- Coordinate Clause (CC):

TREEBANKS

- Treebank:
 - A corpus where each sentence is syntactically annotated
 - Examples:
 - Penn Treebank, Prague Dependency Treebank (Czech), Negra Treebank (German), Susanne (English), Floresta Sintáctica (Linguateca) for Portuguese, ...

EXAMPLE: FLORESTA SINTÁCTICA



SYNTACTIC PARSING

- We call **syntactic analysis/parsing**¹ to the process of obtaining a **syntactic tree/structure** from an input sequence.

¹ We will use these terms interchangeably

SYNTACTIC PARSING

- There are many algorithms to perform syntactic parsing, considering the grammar in use
 - We will study several grammar formalisms (next)
 - There are some approaches that perform syntactic analysis with deep learning methods

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

CONTEXT-FREE GRAMMARS

- Groups of words can behave like single units or sentences; these groups are called “constituents”
 - Examples:
 - Noun Phrases:
 - “the princess”, “John”, “my amazing sister”, ...
- Context-Free Grammars (CFGs):
 - Capture the constituents and their order in sentences

CONTEXT-FREE GRAMMARS

But what
exactly is a
Context-Free
Grammar?



CONTEXT-FREE GRAMMARS (CFG) FORMALISM

- A CFG is a tuple (N, T, S_0, R) in which:
 - N is a set of non-terminal symbols (or tags)
 - Example: n, art, v, NP, VP, ...
 - T is a set of terminal symbols (the language tokens)
 - Example: Maria, love, peace, house, and, ...
 - S_0 is the initial symbol ($S_0 \in N$)
 - R is a set of rules of the form $A \rightarrow a$ where:
 - $A \in N$
 - a is a string of zero or more terminal and non-terminal symbols
 - Example:
 - $NP \rightarrow \text{art } n$

EXAMPLE OF A CONTEXT-FREE GRAMMAR

- $G = (N, T, S_0, R)$,
 - $N = \{S, NP, VP, Pron, Det, Noun, Verb\}$
 - $T = \{I, They, book, João, love, a, the, that\}$
 - $S_0 = S$ (S for sentence)
 - R (note: " $A \rightarrow b \mid c$ " is the same as " $A \rightarrow b$ and $A \rightarrow c$ "):
 - $S \rightarrow NP \ VP$
 - $NP \rightarrow Pron \mid Noun \mid Det \ Noun$
 - $VP \rightarrow Verb \ NP$
 - $Pronoun \rightarrow I \mid They$
 - $Noun \rightarrow book \mid João$
 - $Verb \rightarrow love$
 - $Det \rightarrow a \mid the \mid that$

SYNTACTIC PARSING

- Derivation with CFG: sequence of rule applications in which “A” is rewritten as “a” if there is a rule in the form $A \rightarrow a$
- Language derived by a CFG G:
 - $L(G) = \{w \mid w \text{ is a string of terminal symbols and } S \text{ derives } w\}$

EXAMPLE

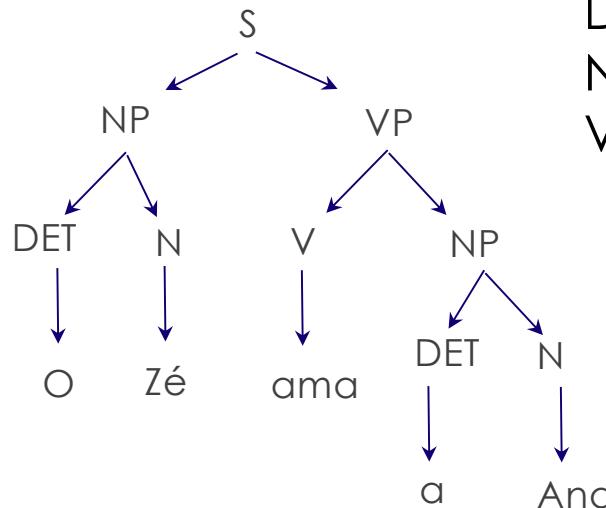
Grammar	Lexicon	POS
S -> NP VP	o, a	DET
NP -> DET N	Zé, Ana	N
VP -> V NP	ama	V

CFC G = (N, T, S₀, R)

- N = {S, NP, VP, DET, N, V}
- T = {o, a, Zé, Ana, ama}
- S₀ = S
- R = {S -> NP VP,
NP -> DET N,
VP -> V NP,
DET -> o | a,
N -> Zé | Ana,
V -> ama}



O Zé
ama a
Ana



Syntactic tree

ACTIVE LEARNING MOMENT



EXERCISE

- Give examples of sentences that belong to $L(G)$, being $G = (N, T, S_0, R)$:
 - $N = \{S, NP, VP, Pron, Det, Noun, Verb\}$
 - $T = \{I, They, book, João, love, a, the, that\}$
 - $S_0 = S$ (S for sentence)
 - $R: \{$
 - $S \rightarrow NP\ VP,$
 - $NP \rightarrow Pron \mid Noun \mid Det\ Noun,$
 - $VP \rightarrow Verb\ NP,$
 - $Pron \rightarrow I \mid They,$
 - $Noun \rightarrow book \mid João,$
 - $Verb \rightarrow love,$
 - $Det \rightarrow a \mid the \mid that\}$
- Use a bottom-up or top-down approach, left-to-right, to show that the sentence “I love that book” belongs to $L(G)$
- Give examples of sentences that do not belong to $L(G)$

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

PROBABILISTIC CFG GRAMMARS FORMALISM

- Probabilistic CFG Grammars
 - Each rule $C \rightarrow a_j$ has a probability associated
 - How to find those probabilities?
 - use a treebank and calculate:
 - A: the number of times $C \rightarrow a_j$ is used
 - B: the number of times the rules of the form $C \rightarrow a_i$ are used
- Then:
- $P(C \rightarrow a_j) [A/B]$

ACTIVE LEARNING MOMENT



EXERCISE

- Consider that in a **treebank**, annotated in terms of the syntactic trees of its sentences, the use of each of the rules of a given grammar is counted:

• $S \rightarrow NP\ VP$	80
• $S \rightarrow Aux\ NP\ VP$	30
• $S \rightarrow VP$	15
• $NP \rightarrow Det\ Nom$	50
• $NP \rightarrow Proper-Noun$	65
• $NP \rightarrow Nom$	15
• $NP \rightarrow Pronoun$	40
• $VP \rightarrow Verb$	40
• $VP \rightarrow Verb\ NP$	40
• $VP \rightarrow Verb\ NP\ NP$	10

What is the probability of the rule $S \rightarrow VP$?

$$\begin{aligned}15/(80+30+15) &= \\15/125 &\end{aligned}$$

PROBABILISTIC CFG GRAMMARS

- Probabilistic CFG Grammars can be used to disambiguate when several parse trees exist
 - The probability of a subtree is the multiplication of the probabilities of its own subtrees; choose the one with the highest probability

ACTIVE LEARNING MOMENT



EXERCISE

- Consider the grammar

$S \rightarrow NP\ VP$	1.0	$NP \rightarrow NP\ PP$	0.4
$PP \rightarrow P\ NP$	1.0	$NP \rightarrow \text{astronomers}$	0.1
$VP \rightarrow V\ NP$	0.7	$NP \rightarrow \text{ears}$	0.18
$VP \rightarrow VP\ PP$	0.3	$NP \rightarrow \text{saw}$	0.04
$P \rightarrow \text{with}$	1.0	$NP \rightarrow \text{stars}$	0.18
$V \rightarrow \text{saw}$	1.0	$NP \rightarrow \text{telescopes}$	0.1

Example from <https://courses.cs.washington.edu/courses/cse590a/09wi/pcfg.pdf>

EXERCISE

- Calculate the two possible syntactic trees for the sentence:

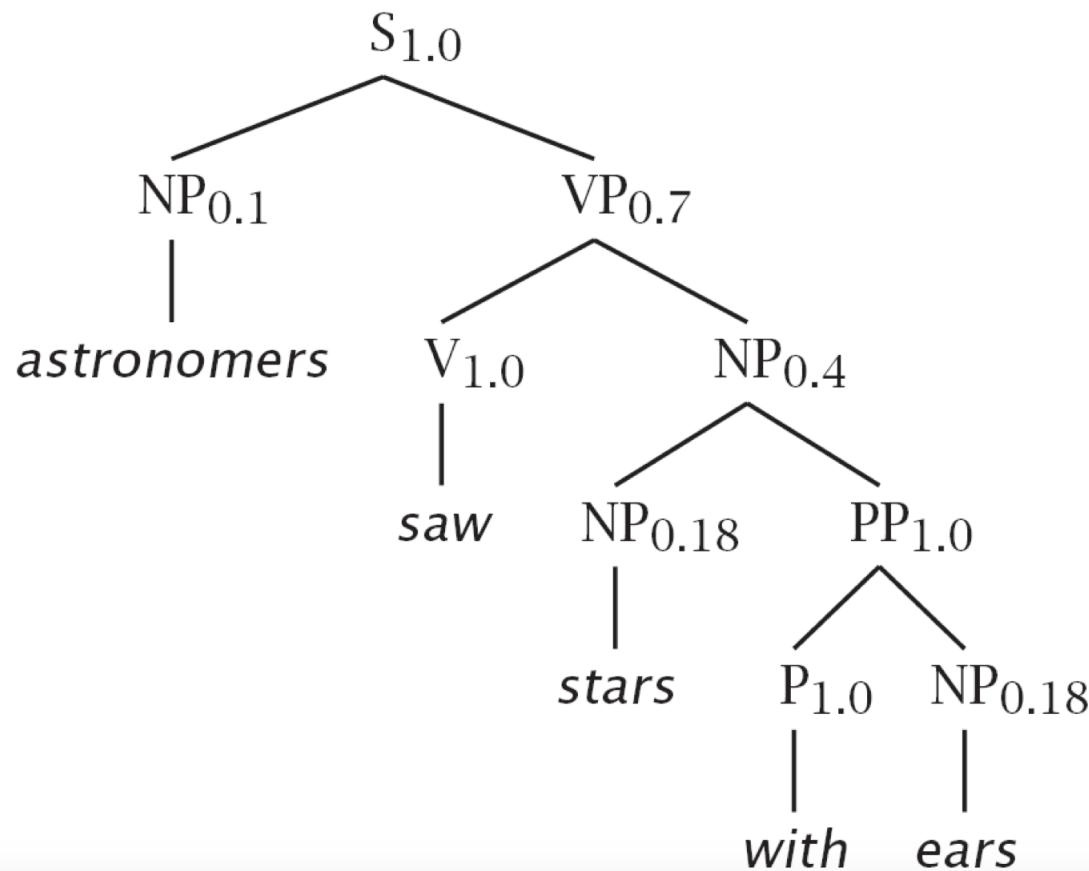
astronomers saw stars with ears

- Then, decide which one is more probable

EXERCISE

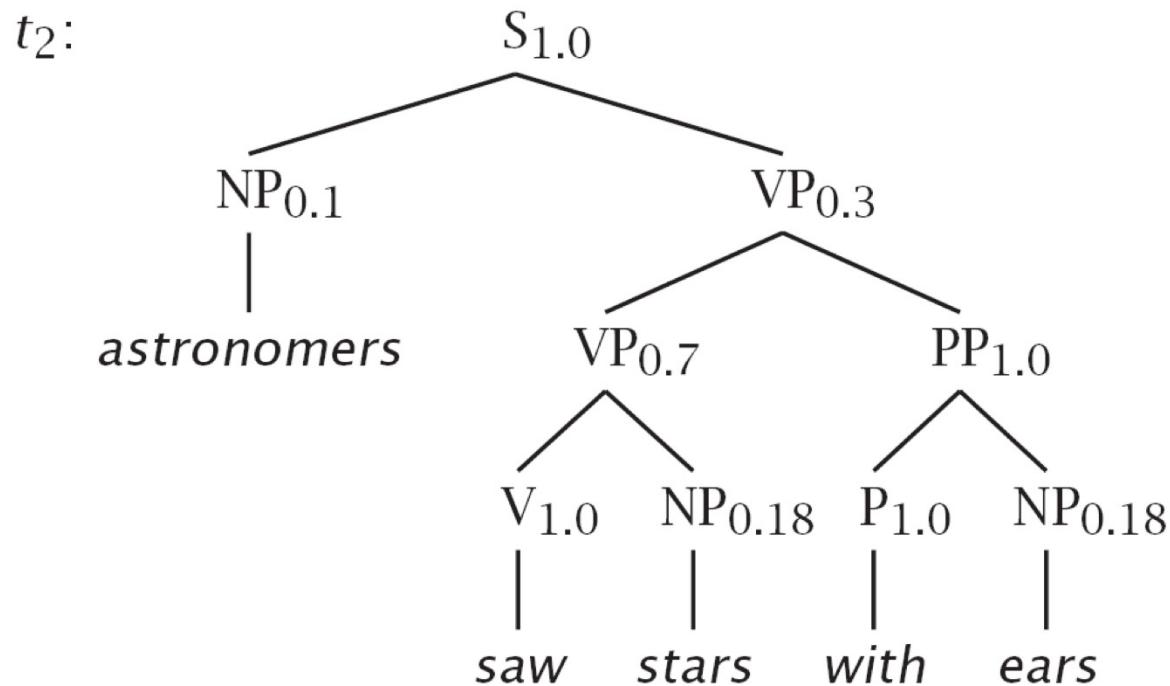
astronomers saw stars with ears

$t_1:$



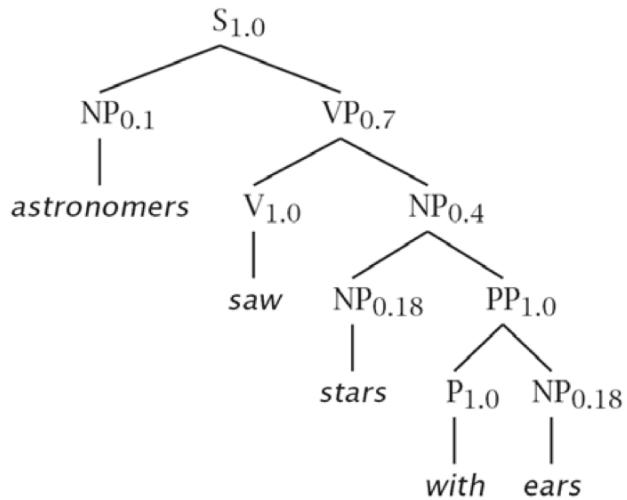
EXAMPLE

astronomers saw stars with ears



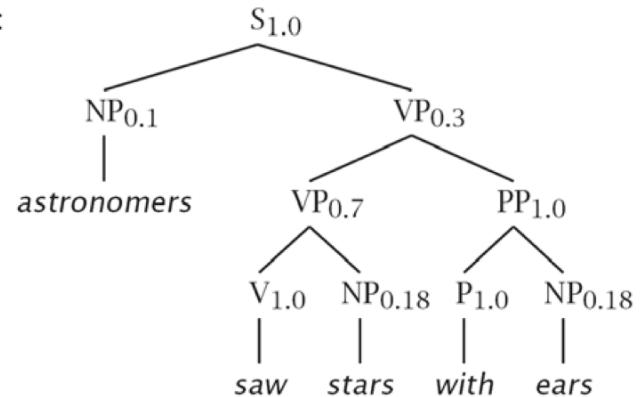
EXAMPLE

$t_1:$



$$\begin{aligned}
 P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\
 &\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
 &= 0.0009072
 \end{aligned}$$

$t_2:$



$$\begin{aligned}
 P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\
 &\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
 &= 0.0006804
 \end{aligned}$$



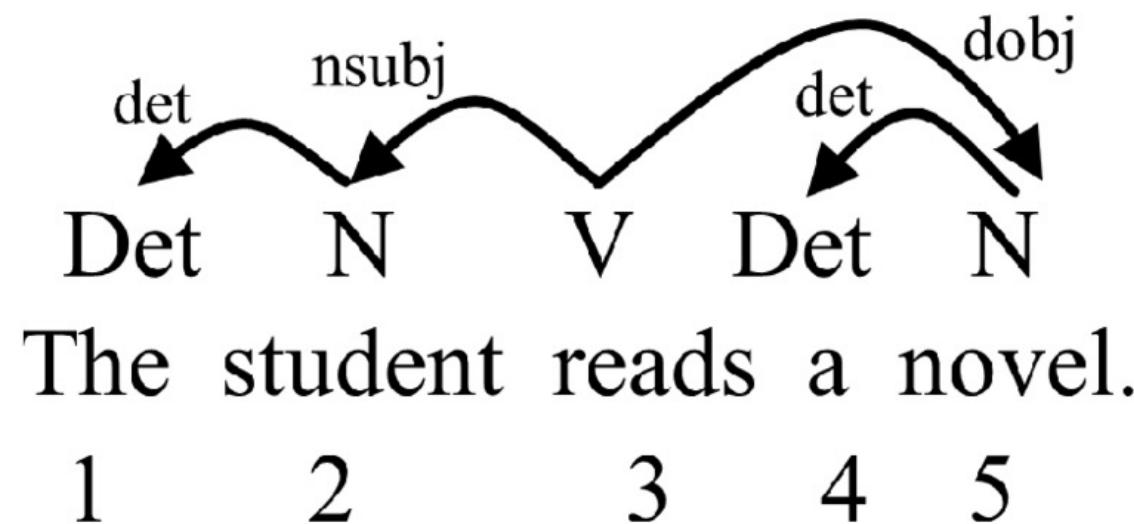
Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

DEPENDENCY GRAMMARS (DG) FORMALISM

- DGs do not use the concept of “constituent”
- A DG has the form $G = (V, A)$, in which:
 - V is a set of vertices (the tokens)
 - A (for arcs) is a set of pairs of vertices
 - Arcs can be labelled
- Each arc represents a (usually grammatical) relation between:
 - The head: role of the central organizing word
 - The dependent: a kind of modifier
- **Derivation with DGs:** sequential application of algorithms that identify and construct the dependency relations among words

EXAMPLE



https://www.researchgate.net/figure/Dependency-structure-of-the-sample-sentence-The-student-reads-a-novel_fig1_332428184

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

EXAMPLE OF A REAL APPLICATION

- EP2LGP5.0: translates from European Portuguese (EP) to the Portuguese Sign Language (LGP)
- Challenge:
 - EP grammar is different from LGP
 - Example:
 - EP: A rainha foi à praia. (The queen went to the beach.)
 - LGP: MULHER REI PRAIA IR (WOMAN KING BEACH GO)



GLOSSES

ACTIVE LEARNING MOMENT



EXERCISE

Portuguese sentences

Preciso ir dormir.

(I need to sleep.)

Queres uma xícara de café?

(Do you want a cup of coffee?)

Você não é uma minoria

(You are not a minority)

Até questionei a minha sanidade

(I even questioned my sanity)

EXERCISE

SIM MEU MULHER SENHOR
(YES MY WOMAN SIR)

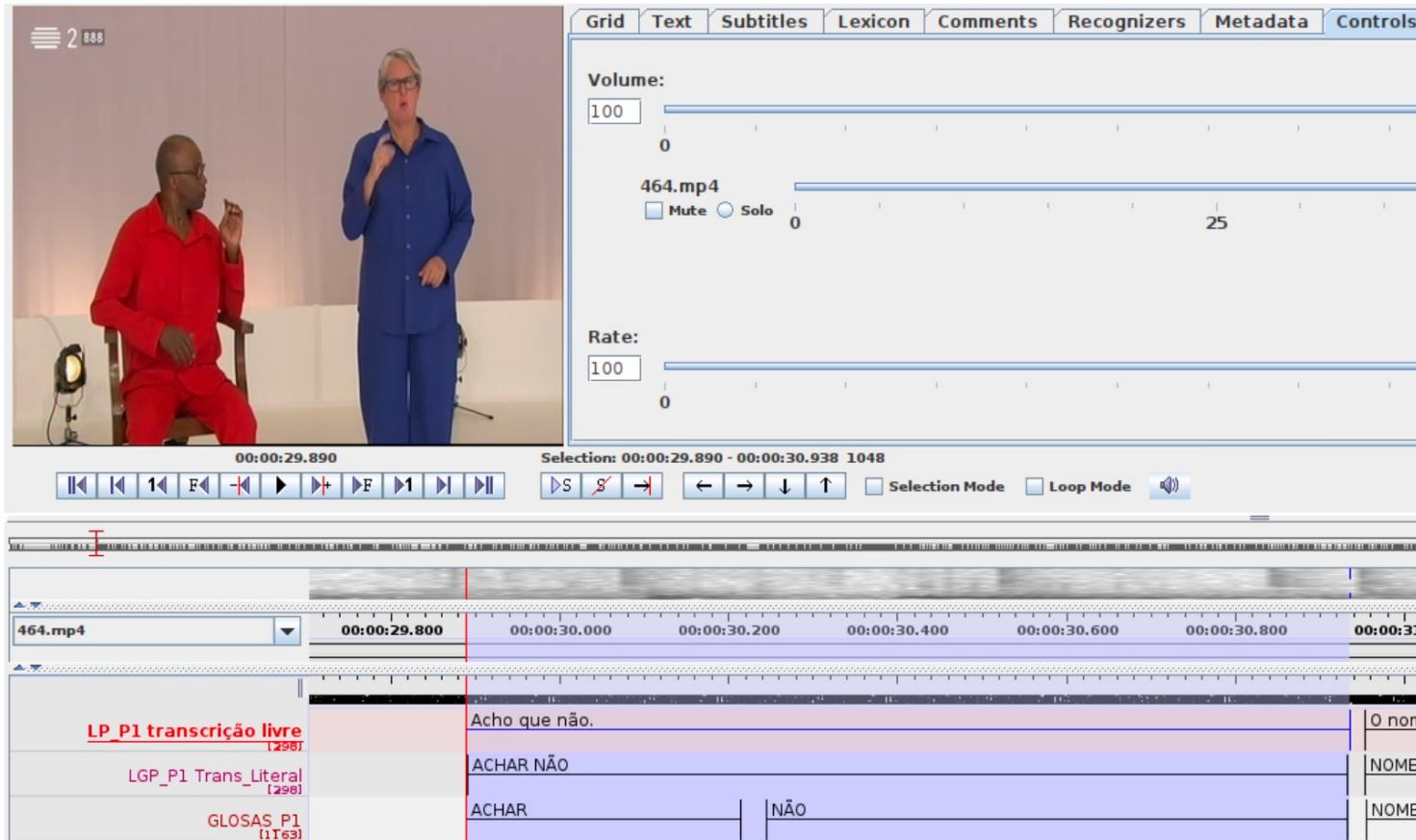
TRISTE MULHER RAPAZ EU ACHAR
(SAD WOMAN BOY I THINK)

192 PARTILHA ELE ATINGIR
(192 SHARE IT REACH)

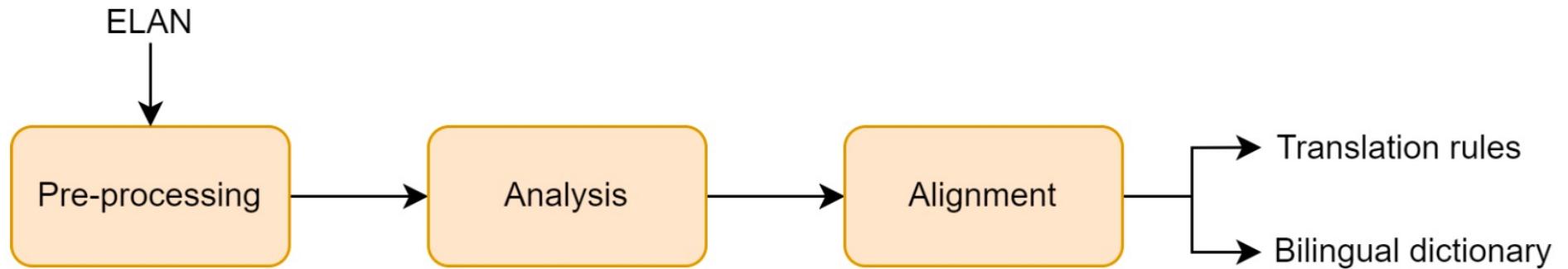
SALARIAIS ALTERNATIVA CORTES
HAVER
(WAGE ALTERNATIVE CUTS THERE
IS)

FROM EP TO LGP (FROM SOUSA 2023)

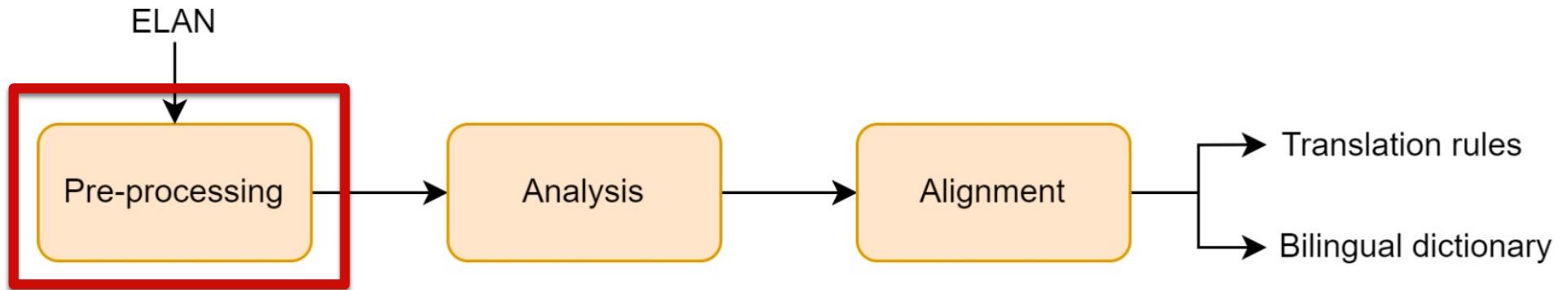
- EP2LGP5.0 takes advantage of an annotated corpus from Católica (with ELAN) – only corpus available between these two languages



FROM EP TO LGP (FROM SOUSA 2023)

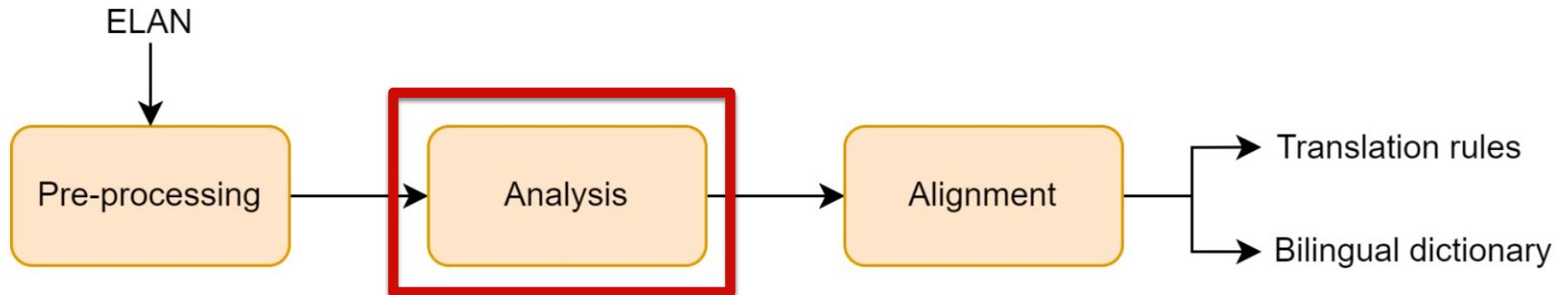


FROM EP TO LGP (FROM SOUSA 2023)

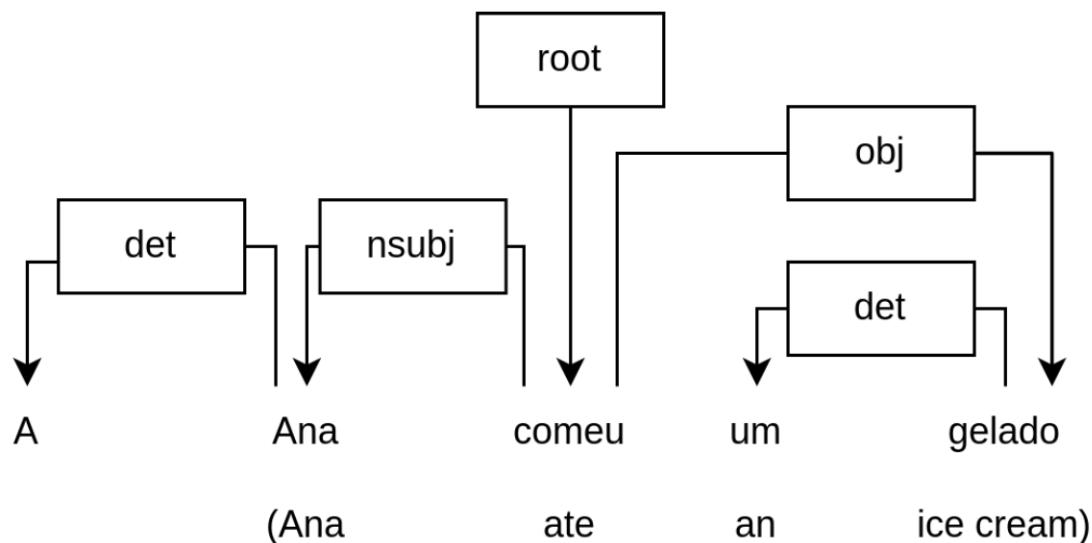


- Extracts the glosses and their grammatical classes, the type of the sentence, ...

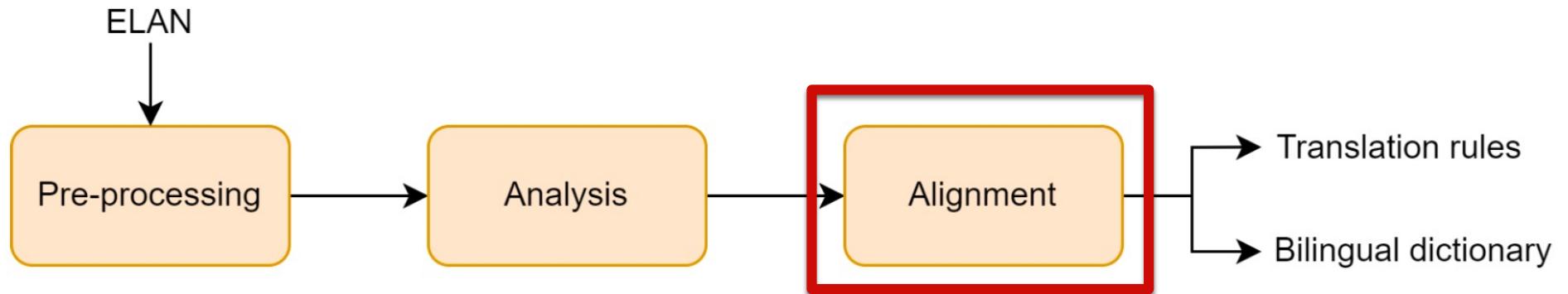
FROM EP TO LGP (FROM SOUSA 2023)



- EP sentences: PoS + Syntactic analysis (dependency relations) + removal of determinants and punctuation



FROM EP TO LGP (FROM SOUSA 2023)



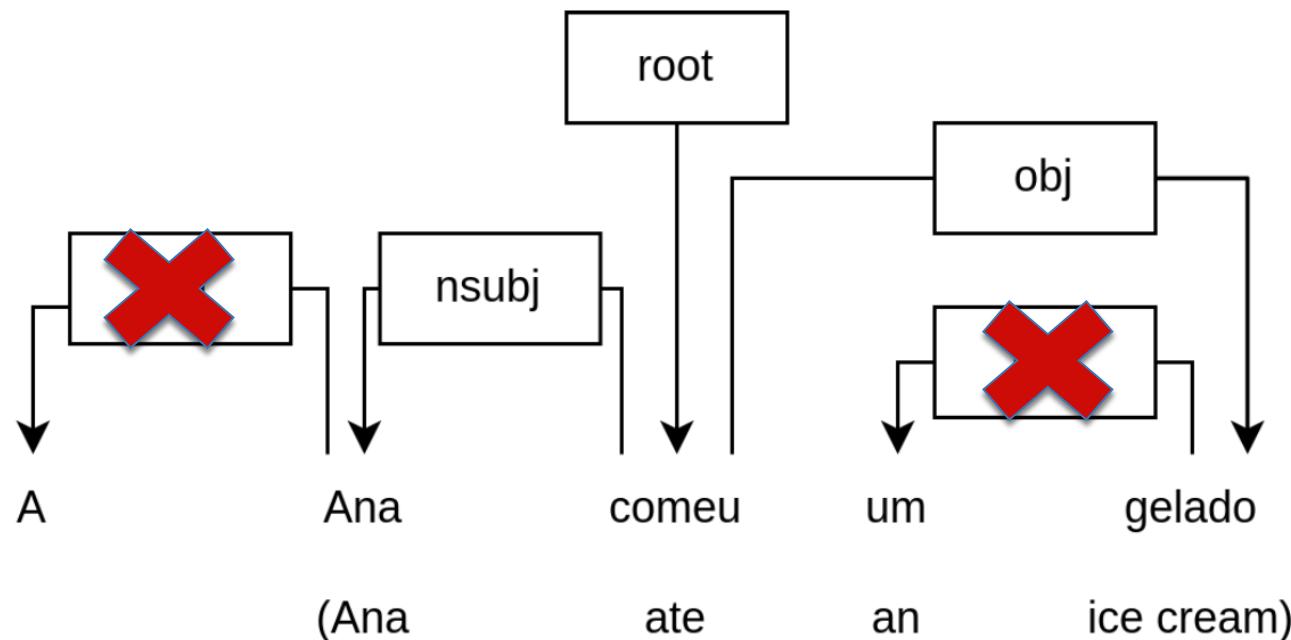
- EP words and LGP glosses are aligned by applying an algorithm based on similarity measures
 - From the alignment, a **bilingual dictionary** and a set of **translation rules** are inferred

FROM EP TO LGP (FROM SOUSA 2023)

- Example of a rule in the [Bilingual Dictionary](#)
 - religião (religion) → IGREJA (CHURCH)
 - houve grande (there was great) → TER – MUITO (HAVE –A–LOT)
- Examples of [Translation Rules](#)
 - Morphosyntactic rules (228 rules)
 - $V_1 N_1 \rightarrow N_1 V_1$
 - General syntactic rules (238 rules)
 - $VP\ NP \rightarrow NP\ VP$

FROM EP TO LGP (FROM SOUSA 2023)

- Running example:
 - A Ana comeu um gelado (Ana ate an ice-cream)



FROM EP TO LGP (FROM SOUSA 2023)

- Running example:
 - A Ana comeu um gelado (Ana ate an ice-cream)

Subject : $N1 \ CAN \rightarrow N1 \ CAN$

Predicate : $V1 \ N2 \ CAN \rightarrow N2 \ V1 \ CAN$

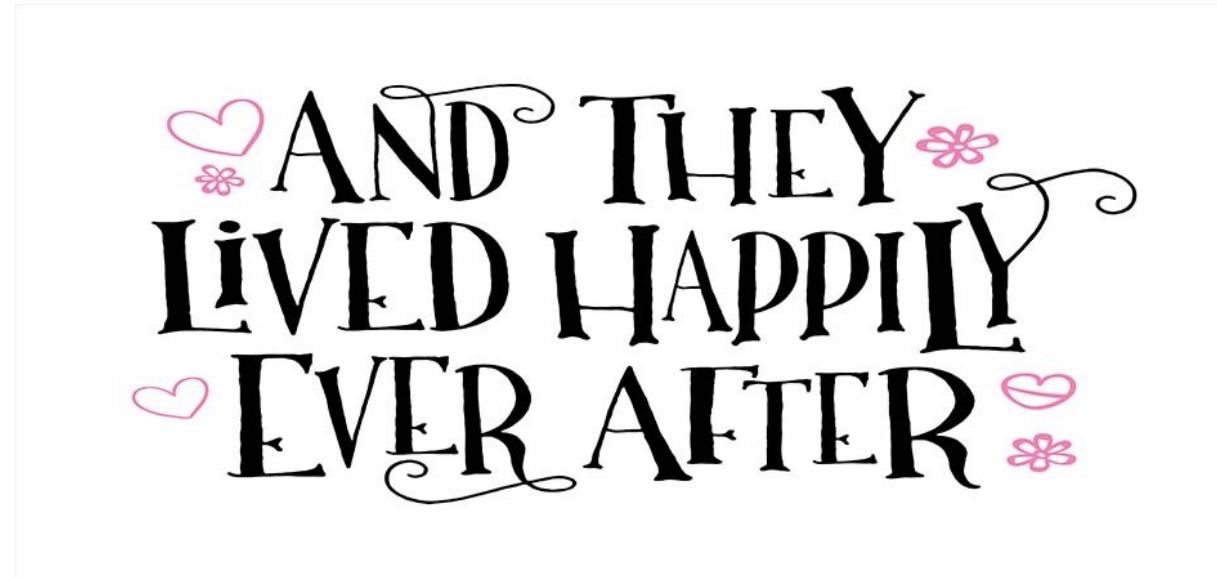
A Ana comeu um gelado. \rightarrow *Ana gelado comeu*

A Ana comeu um gelado. \rightarrow $DT(A - N - A) \ GELADO \ COMER$

- Notes:
 - “CAN” states that the sentence is declarative
 - $DT(A-N-A)$ means that “Ana” (as a proper name) should be fingerspelled

FROM EP TO LGP (FROM SOUSA 2023)

- To the ones that adore Deep Learning:
 - The created rules were used to create a parallel corpus between EP and LGP
 - Deep Learning models were built



Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - An example of a syntactic parser
- Key takeaways
- Suggested readings

LLMs and Syntactic Parsing

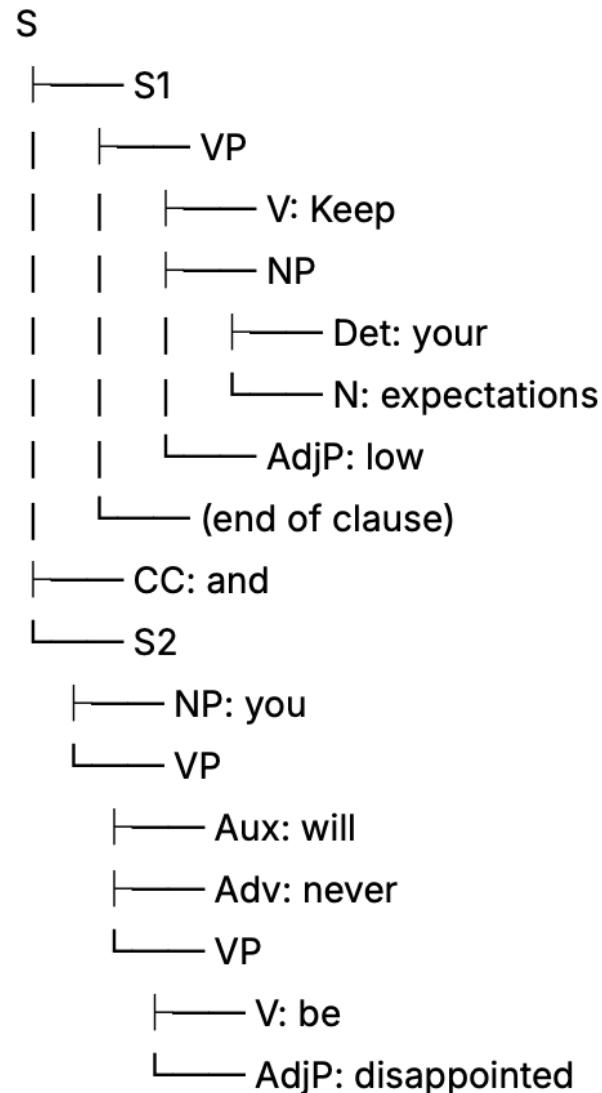
- Task: make a syntactic tree of the sentence:

“keep your expectations low and you'll never be disappointed.”

GEMINI (2.5 FLASH)

```
S (Compound Sentence)
└─ S (Independent Clause 1 / Imperative)
    └─ VP (Verb Phrase)
        └─ V: Keep
        └─ NP (Noun Phrase - Direct Object)
            └─ Det: your
            └─ N: expectations
            └─ AdjP (Adjective Phrase - Object Complement): low
    └─ CONJ: and
    └─ S (Independent Clause 2 / Declarative)
        └─ NP (Noun Phrase - Subject): you
        └─ Aux: 'll (will)
        └─ VP (Verb Phrase)
            └─ Adv: never
            └─ V: be
            └─ AdjP (Adjective Phrase - Predicate Adjective): disappointed
```

GPT-5 Nano



MISTRAL

```
S (Sentence)
  └ ImperativeCl (Imperative Clause)
    └ VP (Verb Phrase)
      └ V (Verb): keep
        └ NP (Noun Phrase)
          └ PossPron (Possessive Pronoun): your
            └ N (Noun): expectations
      └ AP (Adjective Phrase)
        └ Adj (Adjective): low
    └ CoordConj (Coordinate Conjunction): and
    └ S (Sentence)
      └ NP (Noun Phrase)
        └ Pron (Pronoun): you'll
      └ Adv (Adverb): never
      └ VP (Verb Phrase)
        └ Aux (Auxiliary): be
        └ V (Past Participle): disappointed
```

```

S (Compound Sentence)
└── S (Independent Clause 1 / Imperative)
    └── VP (Verb Phrase)
        ├── V: Keep
        └── NP (Noun Phrase - Direct Object)
            ├── Det: your
            └── N: expectations
        └── AdjP (Adjective Phrase - Object Complement): low
    └── CONJ: and
    └── S (Independent Clause 2 / Declarative)
        ├── NP (Noun Phrase - Subject): you
        ├── Aux: 'll (will)
        └── VP (Verb Phrase)
            ├── Adv: never
            ├── V: be
            └── AdjP (Adjective Phrase - Predicate Adjective): disappointed

```

```

S (Sentence)
└── ImperativeCl (Imperative Clause)
    └── VP (Verb Phrase)
        ├── V (Verb): keep
        └── NP (Noun Phrase)
            ├── PossPron (Possessive Pronoun): your
            └── N (Noun): expectations
    └── AP (Adjective Phrase)
        └── Adj (Adjective): low
└── CoordConj (Coordinate Conjunction): and
    └── S (Sentence)
        ├── NP (Noun Phrase)
        │   └── Pron (Pronoun): you'll
        ├── Adv (Adverb): never
        └── VP (Verb Phrase)
            ├── Aux (Auxiliary): be
            └── V (Past Participle): disappointed

```

```

S
└── S1
    └── VP
        ├── V: Keep
        └── NP
            ├── Det: your
            └── N: expectations
        └── AdjP: low
    └── (end of clause)
└── CC: and
└── S2
    ├── NP: you
    └── VP
        ├── Aux: will
        ├── Adv: never
        └── VP
            ├── V: be
            └── AdjP: disappointed

```

Overview

- Learning objectives
- Topics
 - Main concepts
 - Context-free grammars
 - Probabilistic grammars
 - Dependency grammars
 - Example of a real application
 - How good are LLMs in syntactic parsing?
 - [An example of a syntactic parser](#)
- Key takeaways
- Suggested readings

SYNTACTIC PARSING WITH THE CKY ALGORITHM

- The Cocke-Kasami-Younger (CKY) algorithm uses dynamic programming.
- Constraint: grammars must be in the Chomsky Normal Form (CNF).
 - Rules must have one of the following forms:
 - NonTerminal \rightarrow NonTerminal_1 NonTerminal_2
 - NonTerminal \rightarrow terminal

SYNTACTIC PARSING WITH THE CKY ALGORITHM

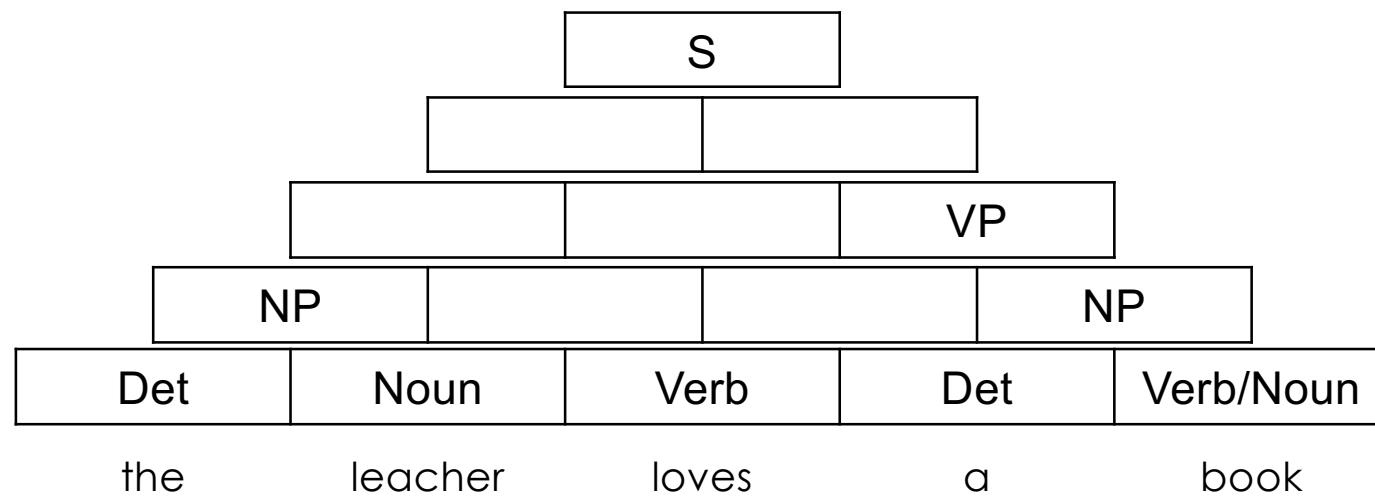
Algorithm 3 CKY

```
j ← 1
while j < n do
    [1, j] = {A : A → wj ∈ R}
    j ++
end while
i ← 2
while i < n do
    j ← 1
    while j < n - i + 1 do
        [i, j] = ∪m=1i-1 {A : A → B C ∈ R, B ∈ [m, j], C ∈ [i - m, j + m]}
        j ++
    end while
    i ++
end while
if S0 ∈ [n, 1] then
    W ∈ L(G)
end if
```

SYNTACTIC PARSING WITH THE CKY ALGORITHM

- Use the CKY algorithm to show that the sentences "the teacher loves a book" $\in L(G)$, and that "the teacher loves a" $\notin L(G)$, being:
- $G = (N, T, S_0, R)$:
 - $N = \{S, NP, VP, Det, Noun, Verb\}$
 - $T = \{\text{book}, \text{João}, \text{love}, \text{a}, \text{the}, \text{that}\}$
 - $S_0 = S$ (S for sentence)
 - $R: \{$
 - $S \rightarrow NP\ VP$
 - $NP \rightarrow Det\ Noun,$
 - $VP \rightarrow Verb\ NP,$
 - $Noun \rightarrow \text{book} \mid \text{table} \mid \text{teacher},$
 - $Verb \rightarrow \text{loves} \mid \text{book}$
 - $Det \rightarrow a \mid \text{the} \mid \text{that}\}$

SYNTACTIC PARSING WITH THE CKY ALGORITHM



KEY TAKEAWAYS

KEY TAKEAWAYS

- Explain the concepts of treebank, constituency grammar, context-free grammar, probabilistic context-free grammar and dependency grammar
- Understand what is the language generated by a CFG
- Apply CKY

SUGGESTED READINGS

READINGS

- Sebenta:
 - Syntax important Natural very Language is
- Jurafsky:
 - Chapter 17 (Context-Free Grammars and...)
 - 17.1-17.3