What is Computer Science?
What is Computer Science?

Systems
What is Computer Science?

Systems

Artificial Intelligence
What is Computer Science?

Systems

Artificial Intelligence

Graphics
What is Computer Science?

- Systems
- Artificial Intelligence
- Graphics
- Security
What is Computer Science?

Systems

Artificial Intelligence

Graphics

Security

Networking
What is Computer Science?

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages
What is Computer Science?

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

...
What is Computer Science?

- Systems
- Artificial Intelligence
- Graphics
- Security
- Networking
- Programming Languages

...
What is Computer Science?

- Systems
- Artificial Intelligence
- Graphics
- Security
- Networking
- Programming Languages
- Computer Vision

...
What is Computer Science?

- Systems
- Artificial Intelligence
- Graphics
- Security
- Networking
- Programming Languages
- Computer Vision
- Planning
...

...
What is Computer Science?

- Systems
- Artificial Intelligence
- Graphics
- Security
- Networking
- Programming Languages
- Computer Vision
- Planning
- Robotics
...
What is Computer Science?

Systems  |  Computer Vision
Artificial Intelligence  |  Planning
Graphics  |  Robotics
Security  |  Natural Language Processing
Networking
Programming Languages
...

...
What is Computer Science?

- Systems
- Artificial Intelligence
- Graphics
- Security
- Networking
- Programming Languages

- Computer Vision
- Planning
- Robotics
- Natural Language Processing
- ...
What is Computer Science?

- Systems
- Artificial Intelligence
- Graphics
- Security
- Networking
- Programming Languages
- Computer Vision
- Planning
- Robotics
- Natural Language Processing
- ...
- ...

...
Machine Translation
Target language corpus gives examples of well-formed sentences

I will get to it later   See you later   He will do it
**Target language corpus gives examples of well-formed sentences**

- I will get to it later
- See you later
- He will do it

**Parallel corpus gives translation examples**

- I will do it gladly
- Yo lo haré de muy buen grado
- You will see later
- Después lo verás
Machine Translation

*Target language corpus gives examples of well-formed sentences*

- I will get to it later
- See you later
- He will do it

*Parallel corpus gives translation examples*

- I will do it gladly
- Yo lo haré de muy buen grado
- You will see later
- Después lo veras

*Machine translation system:*
Machine Translation

Target language corpus gives examples of well-formed sentences

- I will get to it later
- See you later
- He will do it

Parallel corpus gives translation examples

- I will do it gladly
- Yo lo haré de muy buen grado
- You will see later
- Después lo veras

Machine translation system:

Model of translation
Machine Translation

**Target language corpus gives examples of well-formed sentences**

- I will get to it later
- See you later
- He will do it

**Parallel corpus gives translation examples**

- I will do it gladly
- Yo lo haré de muy buen grado
- You will see later
- Después lo veras

**Machine translation system:**

*Source language*
- Yo lo haré después

*Model of translation*

*Target language*
- I will do it later

*Novel sentence*
The Syntactic Structure of Natural Language

Parallel corpus gives translation examples

I will do it gladly

Yo lo haré de muy buen grado

You will see later

Después lo verás

Machine translation system:

Yo lo haré después

Model of translation

I will do it later
The Syntactic Structure of Natural Language

Parallel corpus gives translation examples

Yo lo haré de muy buen grado
I will do it gladly

Después lo veras
You will see later

Yo lo haré después
I will do it later

Parallel corpus gives translation examples

Machine translation system:

Yo lo haré
después
I will do it
later
Parallel corpus gives translation examples

I will do it gladly
Yo lo haré de muy buen grado

You will see later
Después lo verás

Machine translation system:

Yo lo haré  

Model of translation

I will do it  

later
The Syntactic Structure of Natural Language

Parallel corpus gives translation examples

I will do it gladly
Yo lo haré de muy buen grado

You will see later
Después lo verás

Machine translation system:

Yo lo haré después

Model of translation

I will do it later
A Translation System in 20 Minutes
A Translation System in 20 Minutes

Context-free grammars as a model of language generation
A Translation System in 20 Minutes

Context-free grammars as a model of language generation

Large-scale linguistic annotations
A Translation System in 20 Minutes

Context-free grammars as a model of language generation

Large-scale linguistic annotations

Tree transformations
A Translation System in 20 Minutes

Context-free grammars as a model of language generation

Large-scale linguistic annotations

Tree transformations

English
A Translation System in 20 Minutes

Context-free grammars as a model of language generation

Large-scale linguistic annotations

Tree transformations

English → Yoda-speak
A Translation System in 20 Minutes

Context-free grammars as a model of language generation

Large-scale linguistic annotations

Tree transformations

English  \rightarrow  Yoda-speak

Help you, I can!
Yes! Mm!
A Translation System in 20 Minutes

Context-free grammars as a model of language generation

Large-scale linguistic annotations

Tree transformations

English → Yoda-speak

Help you, I can! Yes! Mm!

When 900 years old you reach, look as good, you will not. Hm.
A Context-Free Grammar Describes Generation

"Grammar"
A Context-Free Grammar Describes Generation

"Grammar"

S \rightarrow\ NP\ VP
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
A Context-Free Grammar Describes Generation

"Grammar"

\[
\begin{align*}
S & \rightarrow \text{NP} \ \text{VP} \\
\text{NP} & \rightarrow \text{PRP} \\
\text{VP} & \rightarrow \text{VB}
\end{align*}
\]
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP
A Context-Free Grammar Describes Generation

Grammar:

- **S** → NP VP
- **NP** → PRP
- **VP** → VB
- **VP** → VB PRP

Diagram:

```
S
   / \
NP  VP
```
A Context-Free Grammar Describes Generation

```
S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP

"Grammar"
S  →  NP  VP
NP  →  PRP
VP  →  VB
VP  →  VB  PRP
```
A Context-Free Grammar Describes Generation

```
S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP
```

"Grammar"

```
S  ->  NP  VP
NP  ->  PRP
VP  ->  VB
VP  ->  VB  PRP
```
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
A Context-Free Grammar Describes Generation

```
S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP
PRP -> you
```

"Grammar"
- S -> NP VP
- NP -> PRP
- VP -> VB
- VP -> VB PRP

"Lexicon"
- PRP -> I
- PRP -> you
A Context-Free Grammar Describes Generation

Grammar

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

Lexicon

PRP → I
PRP → you
VB → know
VB → help
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → you
PRP → I
VB → help
VB → know
A Context-Free Grammar Describes Generation

"Grammar"

```
S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP
```

"Lexicon"

```
PRP -> I
PRP -> you
VB -> know
VB -> help
```
A Context-Free Grammar Describes Generation

"Grammar"

\[
S \rightarrow NP \ VP \\
NP \rightarrow PRP \\
VP \rightarrow VB \\
VP \rightarrow VB \ PRP
\]

"Lexicon"

\[
PRP \rightarrow I \\
PRP \rightarrow you \\
VB \rightarrow know \\
VB \rightarrow help
\]
A Context-Free Grammar Describes Generation

Grammar:
- S -> NP VP
- NP -> PRP
- VP -> VB
- VP -> VB PRP
- PRP -> you
- VB -> know
- VB -> help

Lexicon:
- PRP -> I
- PRP -> you
- VB -> know
- VB -> help

Diagram:
- S
  - NP
    - PRP
    - VB
    - PRP → VP
  - VP
- Grammar refinement
A Context-Free Grammar Describes Generation

```
S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP
PRP -> you
VB -> know
VB -> help
```

---

"Grammar"

- S \(\rightarrow\) NP VP
- NP \(\rightarrow\) PRP↓NP
- VP \(\rightarrow\) VB
- VP \(\rightarrow\) VB PRP

"Lexicon"

- PRP \(\rightarrow\) I
- PRP \(\rightarrow\) you
- VB \(\rightarrow\) know
- VB \(\rightarrow\) help
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP
PRP → you
VB → know
VB → help

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help
A Context-Free Grammar Describes Generation

Grammar

\[
S \rightarrow NP \ VP \\
NP \rightarrow PRP \\
PRP \rightarrow \text{you} \\
VP \rightarrow VB \\
VP \rightarrow VB \ PRP \downarrow VP \\
\]

Lexicon

\[
PRP \downarrow NP \rightarrow I \\
PRP \rightarrow \text{you} \\
VB \rightarrow \text{know} \\
VB \rightarrow \text{help} \\
\]
A Context-Free Grammar Describes Generation

```
S -> NP VP
NP -> PRP
PRP
↓
NP -> I
VP -> VB
VP -> VB PRP
PRP -> you
VB -> know
VB -> help

"Grammar"
S  ->  NP  VP
NP  ->  PRP↓NP
VP  ->  VB
VP  ->  VB  PRP↓VP

"Lexicon"
PRP↓NP  ->  I
PRP  ->  you
VB  ->  know
VB  ->  help
PRP↓VP  ->  me
```

Grammar refinement
A Context-Free Grammar Describes Generation

```
Grammar
S  -->  NP  VP
NP  -->  PRP
PRP  -->  I
VP  -->  VB
VP  -->  VB  PRP

Lexicon
PRP  -->  I
PRP  -->  you
VB  -->  know
VB  -->  help
```
A Context-Free Grammar Describes Generation

```
S -> NP VP
NP -> PRP
PRP -> I
VP -> VB
VP -> VB PRP
VP -> MD VP
PRP -> you
VB -> know
VB -> help
```

"Grammar"

```
S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP
VP -> MD VP
```

"Lexicon"

```
PRP -> I
PRP -> you
VB -> know
VB -> help
```
A Context-Free Grammar Describes Generation

"Grammar"

S \rightarrow NP \ VP
NP \rightarrow PRP
PRP \rightarrow I
VP \rightarrow VB
VP \rightarrow VB \ PRP
VP \rightarrow MD \ VP

"Lexicon"

PRP \rightarrow I
PRP \rightarrow you
VB \rightarrow know
VB \rightarrow help
MD \rightarrow can
A Context-Free Grammar Describes Generation

```
S -> NP VP
NP -> PRP
PRP -> I
VP -> VB
VP -> VB PRP
PRP -> you
VP -> MD VP
VB -> know
VB -> help
MD -> can
```

"Grammar"

```
S  ->  NP  VP
NP  ->  PRP
VP  ->  VB
VP  ->  VB  PRP
VP  ->  MD  VP
```

"Lexicon"

```
PRP  ->  I
PRP  ->  you
VB  ->  know
VB  ->  help
MD  ->  can
```
A Context-Free Grammar Describes Generation

**Grammar**

```
S  ->  NP  VP
NP  ->  PRP
PRP  ->  I
VP  ->  VB
VP  ->  VB  PRP
VP  ->  MD  VP
```

**Lexicon**

```
PRP  ->  I
PRP  ->  you
VB  ->  know
VB  ->  help
MD  ->  can
```
A Context-Free Grammar Describes Generation

"Grammar"

S → NP VP
NP → PRP
PRP → I
VP → VB
VP → VB PRP
VP → MD VP
MD → can
VB → help

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help
MD → can
A grammar can be learned from data (demo)
Left-Corner Parsing Builds Trees For Sentences

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help
Left-Corner Parsing Builds Trees For Sentences

"Grammar"

S  ->  NP  VP
NP  ->  PRP
VP  ->  VB
VP  ->  VB  PRP

"Lexicon"

PRP  ->  I
PRP  ->  you
VB  ->  know
VB  ->  help
Left-Corner Parsing Builds Trees For Sentences

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help
Left-Corner Parsing Builds Trees For Sentences

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help

i know you
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)

"Grammar"
S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"
PRP → I
PRP → you
VB → know
VB → help
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)

"Grammar"

S  →  NP  VP
NP  →  PRP
VP  →  VB
VP  →  VB  PRP

"Lexicon"

PRP  →  I
PRP  →  you
VB  →  know
VB  →  help
Left-Corner Parsing Builds Trees For Sentences

```
parse_next(S, 1)
complete(S, (PRP i), 1)
```

**Grammar**
- $S \rightarrow NP \ VP$
- $NP \rightarrow PRP$
- $VP \rightarrow VB$
- $VP \rightarrow VB \ PRP$

**Lexicon**
- PRP $\rightarrow$ I
- PRP $\rightarrow$ you
- VB $\rightarrow$ know
- VB $\rightarrow$ help
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
complete(S, (PRP i), 1)

"Grammar"
S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"
PRP → I
PRP → you
VB → know
VB → help
Left-Corner Parsing Builds Trees For Sentences

\[
\text{parse\_next}(S, 1) \\
\text{complete}(S, (\text{PRP } i), 1) \\
\text{complete}(S, (\text{NP } (\text{PRP } i)), 1)
\]
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
complete(S, (PRP i), 1)
complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)

"Grammar"
S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"
PRP → I
PRP → you
VB → know
VB → help
Left-Corner Parsing Builds Trees For Sentences

```
parse_next(S, 1)
complete(S, (PRP i), 1)
complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
```

"Grammar"

```
S → NP VP
NP → PRP
VP → VB
VP → VB PRP
```

"Lexicon"

```
PRP → I
PRP → you
VB → know
VB → help
```
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
complete(S, (PRP i), 1)
complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help
Left-Corner Parsing Builds Trees For Sentences

```
parse_next(S, 1)
  complete(S, (PRP i), 1)
  complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
  complete(VP, (VB know), 2)
```

**Grammar**

```
S → NP VP
NP → PRP
VP → VB
VP → VB PRP
```

**Lexicon**

```
PRP → I
PRP → you
VB → know
VB → help
```

---

```
S
   NP
      PRP
         VB
            i
             know
               you
```
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
complete(S, (PRP i), 1)
complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
complete(VP, (VB know), 2)

"Grammar"

S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP

"Lexicon"

PRP -> I
PRP -> you
VB -> know
VB -> help

parse_next(S, 1)
complete(S, (PRP i), 1)
complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
complete(VP, (VB know), 2)
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
complete(S, (PRP i), 1)
complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
complete(VP, (VB know), 2)
complete(VP, (VP (VB know)), 2)

"Grammar"

S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP

"Lexicon"

PRP -> I
PRP -> you
VB -> know
VB -> help
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
  complete(S, (PRP i), 1)
  complete(S, (NP (PRP i)), 1)

parse_next(VP, 2)
  complete(VP, (VB know), 2)
  complete(VP, (VP (VB know)), 2)

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help

Left corners
Left-Corner Parsing Builds Trees For Sentences

```
parse_next(S, 1)
  complete(S, (PRP i), 1)
  complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
  complete(VP, (VB know), 2)
  complete(VP, (VP (VB know)), 2)
complete(S, (S ...), 1)
```

**Grammar**

```
S → NP VP
NP → PRP
VP → VB
```

**Lexicon**

```
PRP → I
PRP → you
VB → know
VB → help
```
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
complete(S, (PRP i), 1)
complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
complete(VP, (VB know), 2)

"Grammar"

```
S → NP VP
NP → PRP
VP → VB
VP → VB PRP
```

"Lexicon"

```
PRP → I
PRP → you
VB → know
VB → help
```
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
    complete(S, (PRP i), 1)
    complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
    complete(VP, (VB know), 2)
Left-Corner Parsing Builds Trees For Sentences

```
parse_next(S, 1)
  complete(S, (PRP i), 1)
  complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
  complete(VP, (VB know), 2)
  complete(VP, (VP (VB know) (PRP you)), 2)
```

"Grammar"

```
S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP
```

"Lexicon"

```
PRP -> I
PRP -> you
VB -> know
VB -> help
```
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
    complete(S, (PRP i), 1)
    complete(S, (NP (PRP i)), 1)
parse_next(VP, 2)
    complete(VP, (VB know), 2)
    complete(VP, (VP (VB know) (PRP you)), 2)

"Grammar"

S -> NP VP
NP -> PRP
VP -> VB
VP -> VB PRP

"Lexicon"

PRP -> I
PRP -> you
VB -> know
VB -> help
Left-Corner Parsing Builds Trees For Sentences

parse_next(S, 1)
   complete(S, (PRP i), 1)
   complete(S, (NP (PRP i)), 1)

parse_next(VP, 2)
   complete(VP, (VB know), 2)
   complete(VP, (VP (VB know) (PRP you)), 2)
   complete(S, (S ...), 1)

"Grammar"

S → NP VP
NP → PRP
VP → VB
VP → VB PRP

"Lexicon"

PRP → I
PRP → you
VB → know
VB → help