1. How does the Simplified Lesk Algorithm determine word sense?

Answer:
The **Simplified Lesk Algorithm** is a popular method used for **Word Sense Disambiguation (WSD)** — identifying which meaning of a word is intended in a given context.

Working Principle:

- It is based on the **overlap** between the **dictionary definition (gloss)** of a word's possible senses and the **context words** in the sentence.

Algorithm Steps:

1. Identify the **target word** whose meaning is ambiguous.
2. Retrieve all **possible senses** of the target word from a lexical database such as **WordNet**.
3. For each sense, take the **definition (gloss)** and compare it with the surrounding context words.
4. Count the number of **overlapping words** between the gloss and the sentence context.
5. The sense with the **highest overlap score** is chosen as the correct meaning.

Example:
Sentence: *He went to the bank to deposit money.*
Possible meanings of *bank*:

- (a) River edge
- (b) Financial institution

The words *deposit* and *money* overlap with the *financial institution* definition → hence **bank = financial institution**.

Q16. Find the TF-IDF vector with stop words from the given corpus:

Doc1: “Banana is a fruit”
Doc2: “Apple is a fruit”
Doc3: “Orange and Grape are fruit”

Answer:

Step 1: Vocabulary:
{Banana, Apple, Orange, Grape, is, a, are, fruit}

Step 2: Term Frequency (TF):

Word	D1	D2	D3
Banana	1	0	0
Apple	0	1	0
Orange	0	0	1
Grape	0	0	1
fruit	1	1	1

Step 3: Inverse Document Frequency (IDF):
N = 3 (number of docs)

$$IDF = \log \frac{N}{df}$$

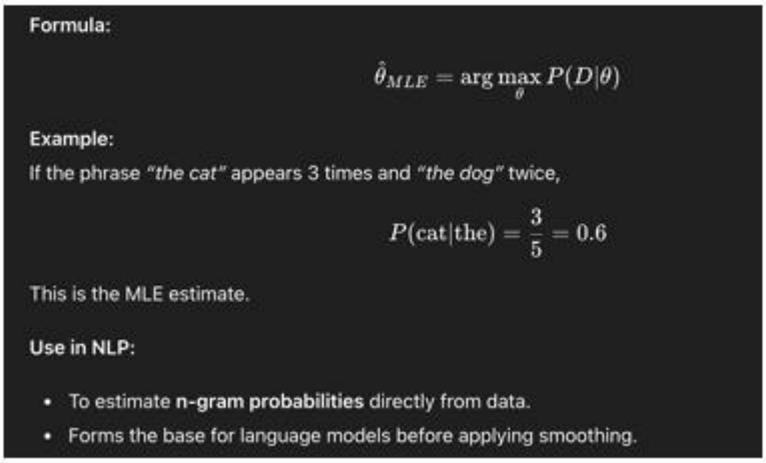
Word	df	IDF
Banana	1	log(3/1)=0.48
Apple	1	0.48
Orange	1	0.48
Grape	1	0.48
fruit	3	0

Step 4: TF–IDF Matrix:

Word	D1	D2	D3
Banana	0.48	0	0
Apple	0	0.48	0
Orange	0	0	0.48
Grape	0	0	0.48
fruit	0	0	0

6. Explain the concept of Maximum Likelihood Estimation (MLE) in your own words.

Answer:
Maximum Likelihood Estimation (MLE) finds the parameters that make the observed data most probable.



Formula:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} P(D|\theta)$$

Example:
If the phrase *"the cat"* appears 3 times and *"the dog"* twice,

$$P(\text{cat}|\text{the}) = \frac{3}{5} = 0.6$$

This is the MLE estimate.

Use in NLP:

- To estimate **n-gram probabilities** directly from data.
- Forms the base for language models before applying smoothing.

4. What is the function of Semantic Role Labeling (SRL) in NLP?

Answer:
Semantic Role Labeling (SRL) identifies and labels the **semantic roles** of sentence constituents — describing *who did what, to whom, when, where, and how*.

It finds **predicate–argument relationships** in text.

Example:
Sentence: *Mary opened the door with a key.*

- Predicate: opened
- Agent (ARG0): Mary
- Patient (ARG1): door
- Instrument (ARG2): key

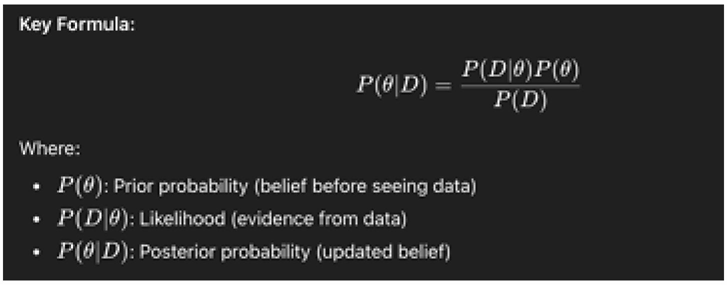
Functions of SRL:

1. Extracts **meaningful relations** between entities.
2. Helps in **text understanding** and **information extraction**.
3. Supports **machine translation**, **summarization**, and **QA systems**.
4. Acts as a step toward **deep semantic understanding** in NLP pipelines.

3. What is the purpose of Bayesian Parameter Estimation in language modeling?

Answer:
Bayesian Parameter Estimation combines prior knowledge and observed data to estimate probabilities in language models.

Key Formula:



Key Formula:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

Where:

- $P(\theta)$: Prior probability (belief before seeing data)
- $P(D|\theta)$: Likelihood (evidence from data)
- $P(\theta|D)$: Posterior probability (updated belief)

Where

$P(\theta)$: Prior probability (belief before seeing data)

$P(D|\theta)$: Likelihood (evidence from data)

$P(\theta|D)$: Posterior probability (updated belief)

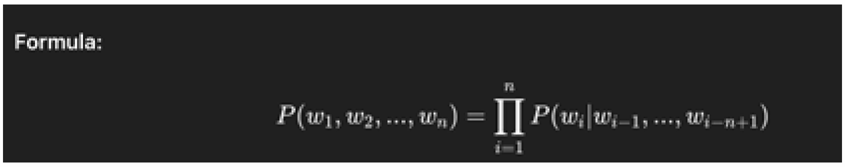
Purpose:

1. Prevents overfitting to small or biased datasets.
2. Deals effectively with **data sparsity**.
3. Gives **probabilistic confidence** in parameter estimates.

Example:
If we rarely see the word "zebra" in a corpus, Bayesian estimation still assigns it a non-zero probability by using prior information.

7. What is the function of n-gram models in predicting word sequences?

Answer:
An **n-gram model** predicts the next word using the previous *(n–1)* words.
It assumes that the probability of a word depends only on its nearby context.



Formula:

$$P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1}, ..., w_{i-n+1})$$

Functions:

1. Estimate probabilities of word sequences.
2. Predict next words in text or speech.
3. Provide context for machine translation and auto-complete.
4. Build statistical understanding of language patterns.

Example:
Sentence: *I love deep learning*.
Trigram:

P(learning|I, love, deep)

12. What do you mean by a Document–Term Matrix? Find the document–term matrix for the given corpus.

Corpus:
Doc1: “I like apple”
Doc2: “I unlike apple”
Doc3: “You like apple”
Doc4: “You unlike apple”

Answer:

Word	D1	D2	D3	D4
I	1	1	0	0
You	0	0	1	1
like	1	0	1	0
unlike	0	1	0	1
apple	1	1	1	1

Q14. What do you mean by the Bag of Words (BoW) model? Find the BoW vector (with stop words) from the given corpus.

Corpus:
Doc1: “I am happy today”
Doc2: “I am unhappy today”
Doc3: “All are happy today”

Answer:

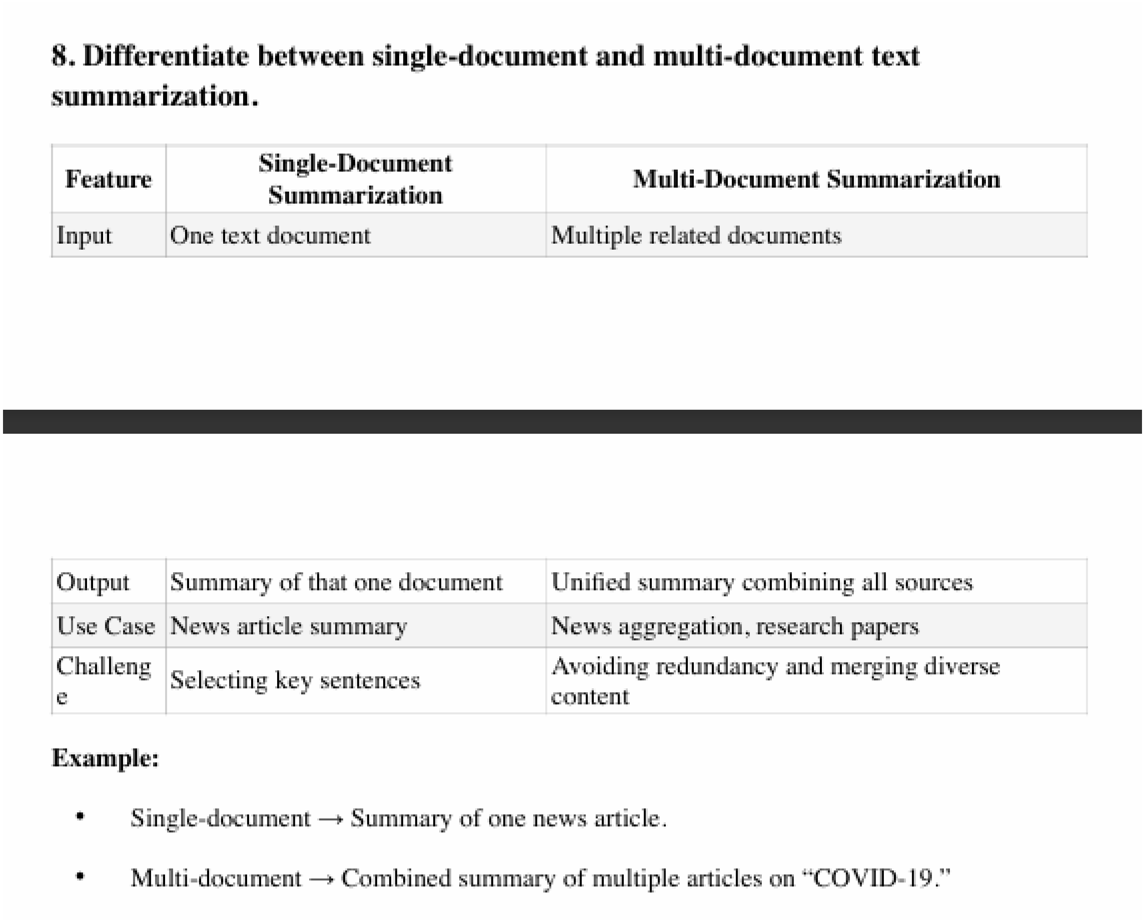
1. Definition:
The **Bag of Words (BoW)** model represents text as a collection (bag) of words, ignoring grammar and order but keeping word frequency.

2. Vocabulary:
{I, am, happy, unhappy, all, are, today}

3. BoW Representation:

Word	Doc1	Doc2	Doc3
I	1	1	0
am	1	1	0
happy	1	0	1
unhappy	0	1	0
all	0	0	1
are	0	0	1
today	1	1	1

BoW converts text into a fixed-length numerical vector, making it suitable for machine learning models in NLP.



8. Differentiate between single-document and multi-document text summarization.

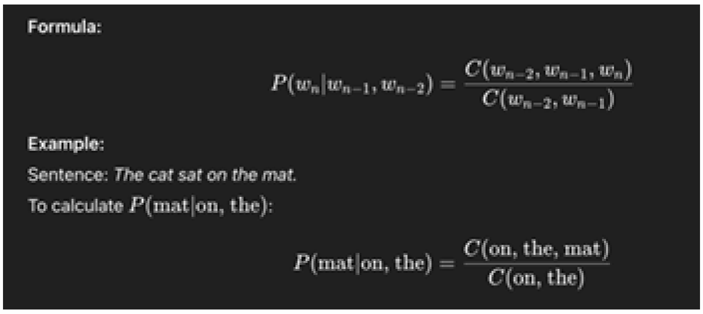
Feature	Single-Document Summarization	Multi-Document Summarization
Input	One text document	Multiple related documents
Output	Summary of that one document	Unified summary combining all sources
Use Case	News article summary	News aggregation, research papers
Challenge	Selecting key sentences	Avoiding redundancy and merging diverse content

Example:

- Single-document → Summary of one news article.
- Multi-document → Combined summary of multiple articles on “COVID-19.”

4. Explain with an example what a Trigram Model is.

Answer:
A **Trigram Model** is an **n-gram language model** that predicts a word using the **previous two words** of context.



Formula:

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

Example:

Sentence: *The cat sat on the mat.*

To calculate $P(\text{mat}|\text{on, the})$:

$$P(\text{mat}|\text{on, the}) = \frac{C(\text{on, the, mat})}{C(\text{on, the})}$$

Advantages:

- Captures longer dependencies than bigram/unigram models.
- Improves fluency in text generation.

1. Why do we use smoothing in NLP?

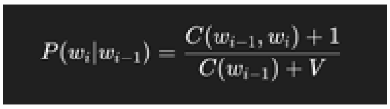
Answer:
In **Language Modeling**, we estimate the probability of word sequences. However, when an **unseen n-gram** (e.g., “cats eat pizza”) appears during testing, its probability becomes **zero**, which can make the entire sentence probability zero.
To solve this, we apply **smoothing**.

Purpose of Smoothing:

1. Avoids zero probability for unseen n-grams.
2. Distributes probability mass from seen words to unseen ones.
3. Improves model generalization and robustness.

Common Smoothing Methods:

- **Add-One (Laplace) Smoothing:**



$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + 1}{C(w_{i-1}) + V}$$

where,V = vocabulary size.

Good–Turing Smoothing: Adjusts frequencies of rare events.

Kneser–Ney Smoothing: Used in advanced n-gram models for best performance.

7. How does GeoQuery serve as a benchmark in semantic parsing research?

Answer:
GeoQuery is a well-known dataset used to **evaluate and compare** semantic parsing systems.

About GeoQuery:

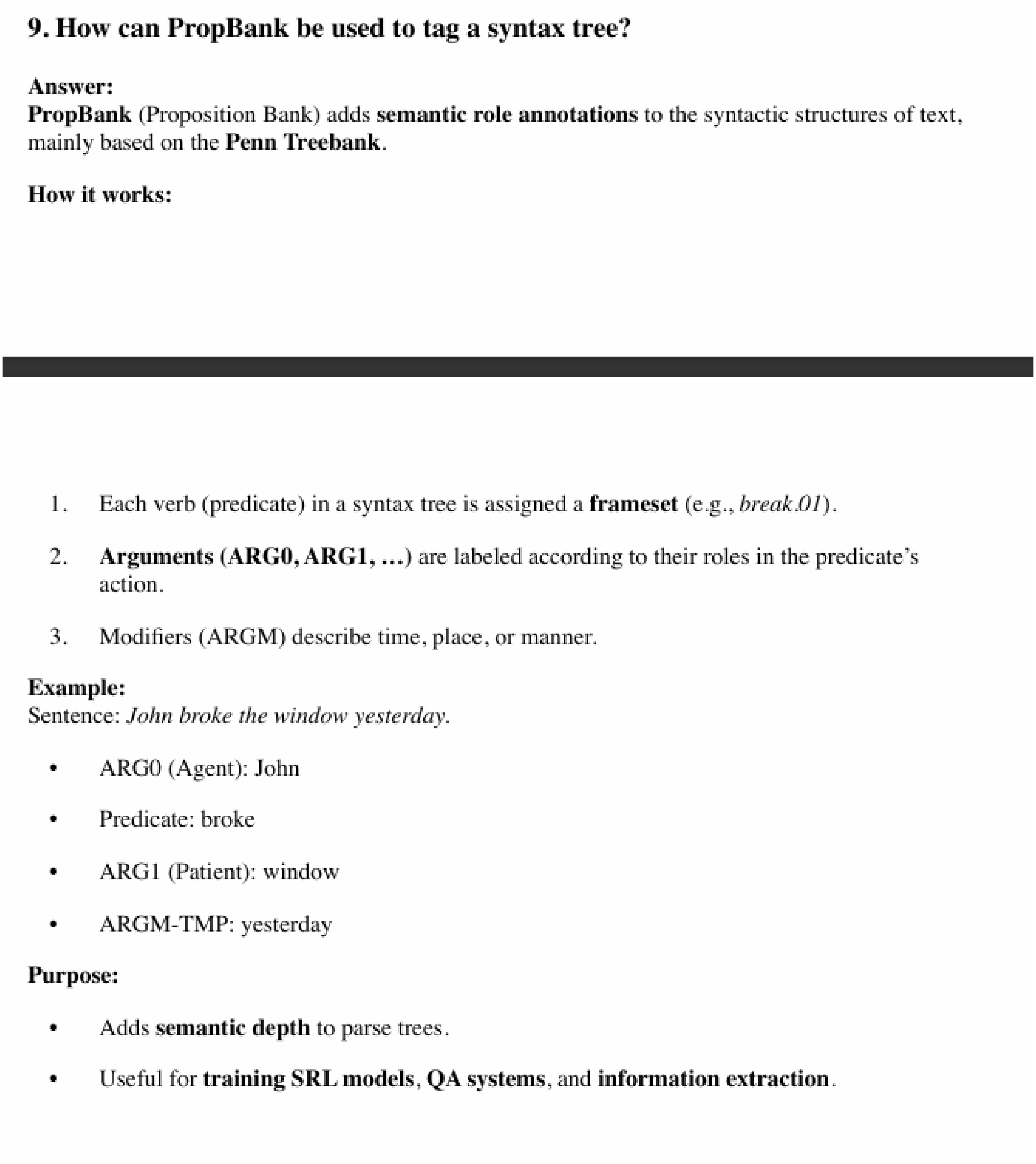
- It contains around **880 natural language questions** about **U.S. geography**, each paired with a **logical query** (usually Prolog-style).

Example:

- Input: *What is the capital of Texas?*
- Output: `answer(capital(Texas))`

Significance:

1. Provides a **standard benchmark** for measuring semantic parser accuracy.
2. Encourages consistent **evaluation and comparison** across systems.
3. Helps test the ability of parsers to map **natural language → logic forms**.



9. How can PropBank be used to tag a syntax tree?

Answer:
PropBank (Proposition Bank) adds **semantic role annotations** to the syntactic structures of text, mainly based on the **Penn Treebank**.

How it works:

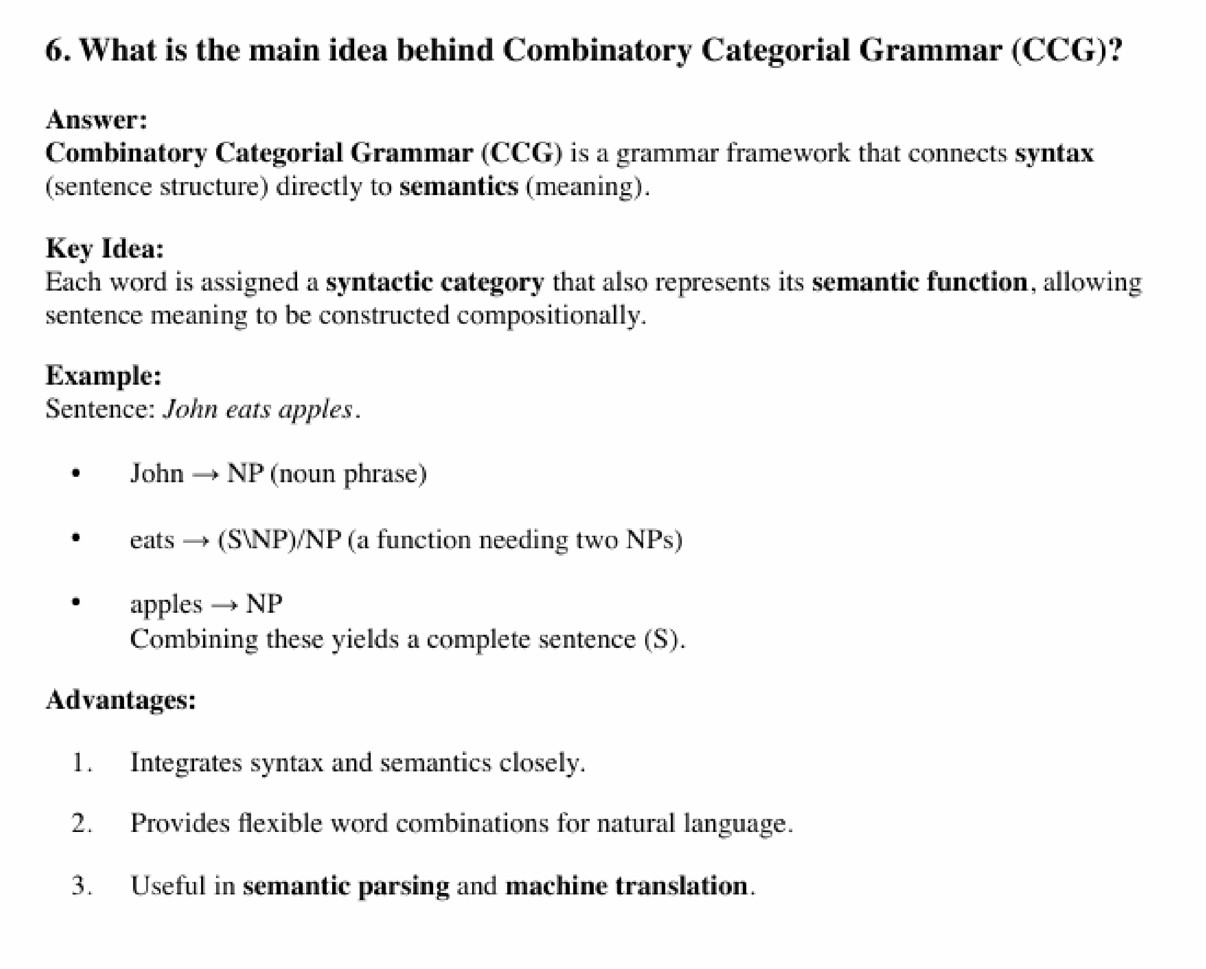
1. Each verb (predicate) in a syntax tree is assigned a **frameset** (e.g., *break.01*).
2. **Arguments (ARG0, ARG1, …)** are labeled according to their roles in the predicate's action.
3. Modifiers (ARGM) describe time, place, or manner.

Example:
Sentence: *John broke the window yesterday.*

- ARG0 (Agent): John
- Predicate: broke
- ARG1 (Patient): window
- ARGM-TMP: yesterday

Purpose:

- Adds **semantic depth** to parse trees.
- Useful for **training SRL models**, **QA systems**, and **information extraction**.



6. What is the main idea behind Combinatory Categorial Grammar (CCG)?

Answer:
Combinatory Categorial Grammar (CCG) is a grammar framework that connects **syntax** (sentence structure) directly to **semantics** (meaning).

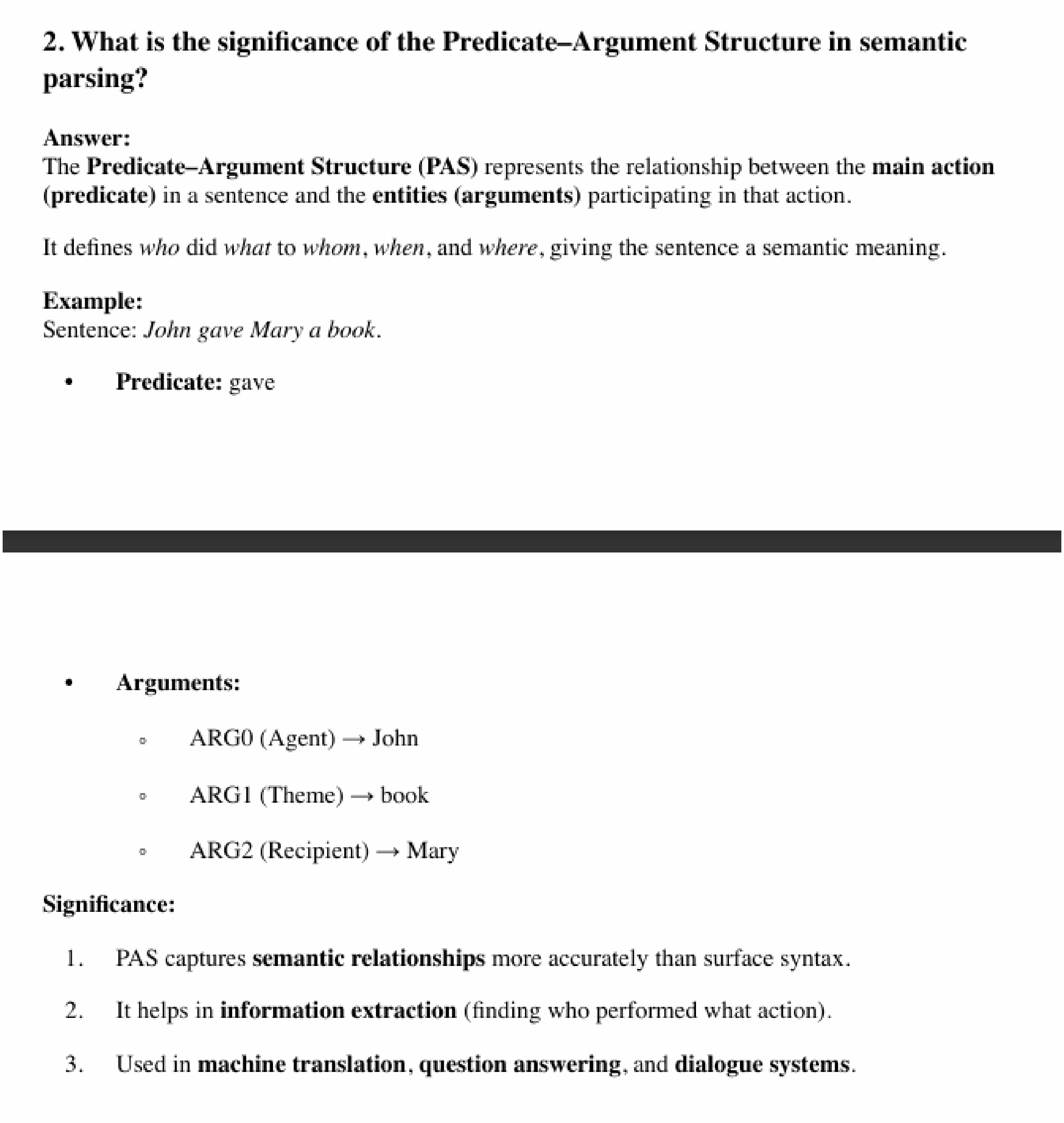
Key Idea:
Each word is assigned a **syntactic category** that also represents its **semantic function**, allowing sentence meaning to be constructed compositionally.

Example:
Sentence: *John eats apples*.

- John → NP (noun phrase)
- eats → (S\NP)/NP (a function needing two NPs)
- apples → NP
  Combining these yields a complete sentence (S).

Advantages:

1. Integrates syntax and semantics closely.
2. Provides flexible word combinations for natural language.
3. Useful in **semantic parsing** and **machine translation**.



2. What is the significance of the Predicate–Argument Structure in semantic parsing?

Answer:
The **Predicate–Argument Structure (PAS)** represents the relationship between the **main action (predicate)** in a sentence and the **entities (arguments)** participating in that action.

It defines *who* did *what* to *whom*, *when*, and *where*, giving the sentence a semantic meaning.

Example:
Sentence: *John gave Mary a book.*

- **Predicate:** gave
- **Arguments:**
  - ARG0 (Agent) → John
  - ARG1 (Theme) → book
  - ARG2 (Recipient) → Mary

Significance:

1. PAS captures **semantic relationships** more accurately than surface syntax.
2. It helps in **information extraction** (finding who performed what action).
3. Used in **machine translation**, **question answering**, and **dialogue systems**.