<u>Cohesion: (Discourse Cohesion)</u>

In **Natural Language Processing (NLP)**, **cohesion** refers to the way different elements of a text are connected to ensure clarity and logical flow. It helps in making a text understandable by linking words, phrases, and sentences together.

Types of Cohesion in NLP:

1. **Lexical Cohesion** – The repetition of words or the use of synonyms, antonyms, or related terms to maintain consistency.
    - *Example*: "The **dog** barked. The **animal** seemed restless."
2. **Grammatical Cohesion** – The use of pronouns, conjunctions, and determiners to maintain reference across sentences.
    - *Example*: "John loves coding. **He** spends hours writing scripts."
3. **Reference Cohesion** – The use of pronouns or definite articles to refer back to previously mentioned entities.
    - *Example*: "Sarah bought a new book. **The book** was very interesting."
4. **Substitution & Ellipsis** – Replacing or omitting words to avoid redundancy.
    - *Example*:
        - Substitution: "I like apples, and **so does** Mark."
        - Ellipsis: "I ordered pizza, and John (ordered) **too**."
5. **Conjunctive Cohesion** – Using conjunctions and discourse markers to show relationships between ideas.
    - *Example*: "She was tired, **so** she went to bed."

Importance of Cohesion in NLP:

- Improves **text understanding** for models like ChatGPT.
- Helps in **text summarization** and **question-answering** tasks.
- Enhances **coherence in machine-generated text**.
- Used in **coreference resolution** (determining which words refer to the same entity).

Reference Resolution

Reference resolution is the process of identifying what a word or phrase refers to in a given text. It is essential for understanding **pronouns, definite descriptions, and anaphoric expressions** in NLP tasks like text summarization, question answering, and machine translation.

Types of Reference Resolution:

1. **Coreference Resolution** – Identifying multiple expressions that refer to the same entity.

        - **Example:**
            - *"John bought a book. **He** loves reading."*
            - (**He** refers to **John**)
2. **Anaphora Resolution** – Resolving pronouns or noun phrases that refer to something mentioned earlier.
    - **Example:**
        - *"I saw a dog. **The animal** was barking."*
        - (**The animal** refers to **dog**)
3. **Cataphora Resolution** – When a pronoun or reference appears before the entity it refers to.
    - **Example:**
        - *"Although **he** was late, Tom still joined the meeting."*
        - (**He** refers to **Tom**)
4. **Bridging Resolution** – Inferring relationships between entities that are not explicitly stated.
    - **Example:**
        - *"I bought a book. **The cover** looks nice."*
        - (**The cover** is part of **the book**, inferred through context.)

Why is Reference Resolution Important?

- **Improves Text Understanding** – Helps NLP models comprehend context.
- **Aids in Summarization & Question Answering** – Resolves references for accurate responses.
- **Enhances Chatbots & Virtual Assistants** – Ensures coherent and meaningful conversations.
- **Useful in Machine Translation** – Maintains correct reference mapping across languages.

N-Gram Models

An **n-gram model** is a **probabilistic language model** that predicts the next word in a sequence based on the previous n−1 words. It is widely used in **speech recognition, machine translation, text generation, and spelling correction**.

What is an N-Gram?

An **n-gram** is a contiguous sequence of n words from a given text.

- **Unigram (1-gram)** → Single word: `"hello"`
- **Bigram (2-gram)** → Two words: `"hello world"`
- **Trigram (3-gram)** → Three words: `"hello world today"`
- **4-gram, 5-gram, etc.** → Longer sequences

Example Sentence:

"I love natural language processing."

N-Gram Type	Example
Unigrams	`"I"`, `"love"`, `"natural"`, `"language"`, `"processing"`
Bigrams	`"I love"`, `"love natural"`, `"natural language"`, `"language processing"`
Trigrams	`"I love natural"`, `"love natural language"`, `"natural language processing"`

- N-gram models use **conditional probability** to predict the next word based on the previous words.

$$P(w_n|w_{n-1}, w_{n-2}, ..., w_1) = \frac{C(w_1, w_2, ..., w_n)}{C(w_1, w_2, ..., w_{n-1})}$$

Where:

- $P(w_n|w_{n-1}, w_{n-2}, ..., w_1)$ is the probability of the next word given previous words.
- $C(w_1, w_2, ..., w_n)$ is the count of the sequence in the dataset.
- $C(w_1, w_2, ..., w_{n-1})$ is the count of the prefix sequence.

Example:

Given a bigram model, the probability of the phrase `"I love"` might be calculated as:

$$P(\text{"love"}|\text{"I"}) = \frac{C(\text{"I love"})}{C(\text{"I"})}$$

Language Model Evaluation

Language model evaluation is the process of measuring how well a language model performs on various NLP tasks. It helps determine the effectiveness, accuracy, and efficiency of the model in generating, understanding, and predicting text.

Metrics for Evaluating Language Models

A. Perplexity (PPL) – For Probability-Based Models

- Measures how **well a model predicts a sample of text**.
- **Lower perplexity** = **better model** (more confident predictions).

$$PPL(W) = P(w_1, w_2, ..., w_N)^{-\frac{1}{N}}$$

Example:

- **Model A**: PPL = 30 (better)
- **Model B**: PPL = 100 (worse)

B. BLEU (Bilingual Evaluation Understudy) – For Machine Translation

- Compares machine-generated text to **human reference translations**.
- Measures **n-gram precision** with a brevity penalty to avoid short outputs.

Example:
If the reference is *"The cat sits on the mat."* and the model predicts *"A cat is on the mat."*, BLEU will score based on overlapping words.

Why is Evaluation Important?

- **Ensures model reliability** before deployment.
- **Compares models** (e.g., GPT vs. BERT vs. LLaMA).
- **Identifies biases** and errors in NLP systems

Parameter Estimation

- Parameter estimation is the process of determining the optimal values of parameters in a statistical or machine learning model.

- It helps in computing probabilities for language models, such as **n-gram models, Hidden Markov Models (HMMs), and neural networks**.

Methods of Parameter Estimation

There are several ways to estimate parameters in NLP models:

A. Maximum Likelihood Estimation (MLE)

- Estimates parameters by **maximizing the probability** of observed data.
- Common in **n-gram language models, HMMs**.

Formula for MLE in a bigram model:

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

where:

- $C(w_{n-1}, w_n)$ = Count of bigram occurrences.
- $C(w_{n-1})$ = Count of unigram occurrences.

🔹 **Issue:** MLE assigns **zero probability** to unseen events.

B. Laplace (Add-One) Smoothing

- Avoids zero probabilities by adding **+1 to all counts**.
- Used in **n-gram models** to improve generalization

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

where V is the vocabulary size.

C. Bayesian Estimation (Maximum A Posteriori - MAP)

- Improves MLE by incorporating **prior knowledge** (Bayesian inference).
- Used in **Naïve Bayes classifiers** in text classification.

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

where:

- $P(D|\theta)$ = Likelihood from data.
- $P(\theta)$ = Prior knowledge.

D. Expectation-Maximization (EM) Algorithm

- Used for models with **hidden variables**, like **HMMs, topic models (LDA)**.
- Alternates between:
  - **Expectation (E-step)** → Estimate missing data.
  - **Maximization (M-step)** → Optimize parameters.

Example: Estimating Bigram Probabilities

Suppose we have the following text:
📜 **"The cat sat. The cat ran."**

- **Bigram Counts:**
  - `C(the, cat) = 2`
  - `C(cat, sat) = 1`, `C(cat, ran) = 1`
  - `C(the) = 2`, `C(cat) = 2`

Using **MLE:**

$$P(cat|the) = \frac{C(the, cat)}{C(the)} = \frac{2}{2} = 1$$

$$P(sat|cat) = \frac{C(cat, sat)}{C(cat)} = \frac{1}{2} = 0.5$$

$$P(ran|cat) = \frac{C(cat, ran)}{C(cat)} = \frac{1}{2} = 0.5$$

📌 **Improvement:** If `"cat jumped"` is unseen, MLE gives $P(jumped|cat) = 0$.
We can apply **Laplace smoothing** to assign a small probability instead.

<u>Types of Language Models:</u>

1. Class-Based Language Model

A **Class-Based Language Model (CBLM)** is an extension of **n-gram models**, where words are grouped into **classes** based on their **semantic or syntactic similarity**. Instead of predicting a word **directly**, the model first predicts the **class** and then the **word within that class**.

Formula

The probability of a word sequence w1,w2,...,wnw_1, w_2, ..., w_nw1,w2,...,wn is given by:

$$P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(C_i | w_{i-1}, w_{i-2}, ...) P(w_i | C_i)$$

where:

- C_i is the class of word w_i.
- $P(C_i | w_{i-1}, w_{i-2}, ...)$ is the probability of the class given the context.
- $P(w_i | C_i)$ is the probability of the word given its class.

Example

- Words like **"cat," "dog," "elephant"** belong to the **"animals"** class.
- Instead of learning probabilities for **individual words**, the model learns:
  - $P(C = \text{Animals} | \text{previous words})$
  - $P(w = \text{dog} | C = \text{Animals})$

This **reduces sparsity** and improves **generalization** for rare words.

2. Length-Based Language Model

A **Length Language Model (LLM)** predicts the probability of a sentence based on its **length distribution**. It is useful for **speech recognition** and **machine translation**, where sentence length is **important**.

Formula

The probability of a sentence SSS of length LLL is:

$$P(S) = P(L)P(S|L)$$

where:

- $P(L)$ is the probability of a sentence having length L.
- $P(S|L)$ is the probability of the sentence given its length.

Example

- In **speech recognition**, shorter phrases like **"Yes"** should be predicted with **higher probability** than long phrases like **"Yes, I completely agree with that statement."**
- In **machine translation**, a **long source sentence** should correspond to a **long target sentence**.

3. Discriminative Language Model

Unlike **generative models**, a **Discriminative Language Model (DLM)** learns to **differentiate between correct and incorrect sequences** rather than modeling their probability distribution.

Formula

Discriminative models use a scoring function:

$$P(S) = \frac{\exp(f(S))}{\sum_{S'} \exp(f(S'))}$$

where:

- $f(S)$ is a feature function that scores sentence S.
- The denominator normalizes over all possible sentences.

Example

- Used in **machine translation** and **speech recognition** to rerank outputs.
- If a model generates:
  1. "I am going to school."
  2. "Going school am I to."
  - The **DLM assigns a higher score** to the first sentence.

4. Syntax-Based Language Model

These models incorporate **syntactic structures** into language modeling, often using **probabilistic parsing**.

A. Structured Language Models

- Use **syntactic trees** to model sentence structures.
- **Example: Probabilistic Context-Free Grammar (PCFG)**
  - Uses probabilities for **grammar rules**.

$$P(S) = \prod P(R_i)$$

where R_i are the **rules used** in the parse tree.

Example

- PCFG rule probabilities:
  - $P(\text{S} \rightarrow \text{NP VP}) = 0.9$
  - $P(\text{VP} \rightarrow \text{V NP}) = 0.7$
- Generates: **"The cat sleeps"** with probability = 0.9×0.7.

B. Almost Parsing Models

- Learn syntax **implicitly** rather than explicitly using parse trees.

- Example: **Recurrent Neural Networks with Attention**.

5. Maximum Entropy (MaxEnt) Language Model

A **MaxEnt model** predicts the probability of a word **without making independence assumptions** like n-gram models.

Formula

$$P(w|h) = \frac{\exp(\sum_i \lambda_i f_i(w,h))}{Z(h)}$$

where:

- λ_i are parameters (learned from data).
- $f_i(w,h)$ are **features** (e.g., POS tags, previous words).
- $Z(h)$ is a normalizing term.

Example

- Used in **speech recognition** and **text classification**.
- Features include **POS tags, previous words, and syntax rules**.

6. Factored Language Model (FLM)

A **Factored Language Model (FLM)** decomposes words into **multiple factors** such as **word form, POS tags, and morphology**.

Formula

$$P(w_i|h) = \prod_k P(F_{i,k}|h)$$

where $F_{i,k}$ are factors like **POS tags or subword units**.

Example

- "Running" → (Word: "Running", POS: Verb, Tense: Present).
- Used in **morphologically rich languages** (e.g., German, Finnish).

7. Neural Network Language Model (NNLM)

Neural network-based models use **embeddings** instead of **discrete word probabilities**.

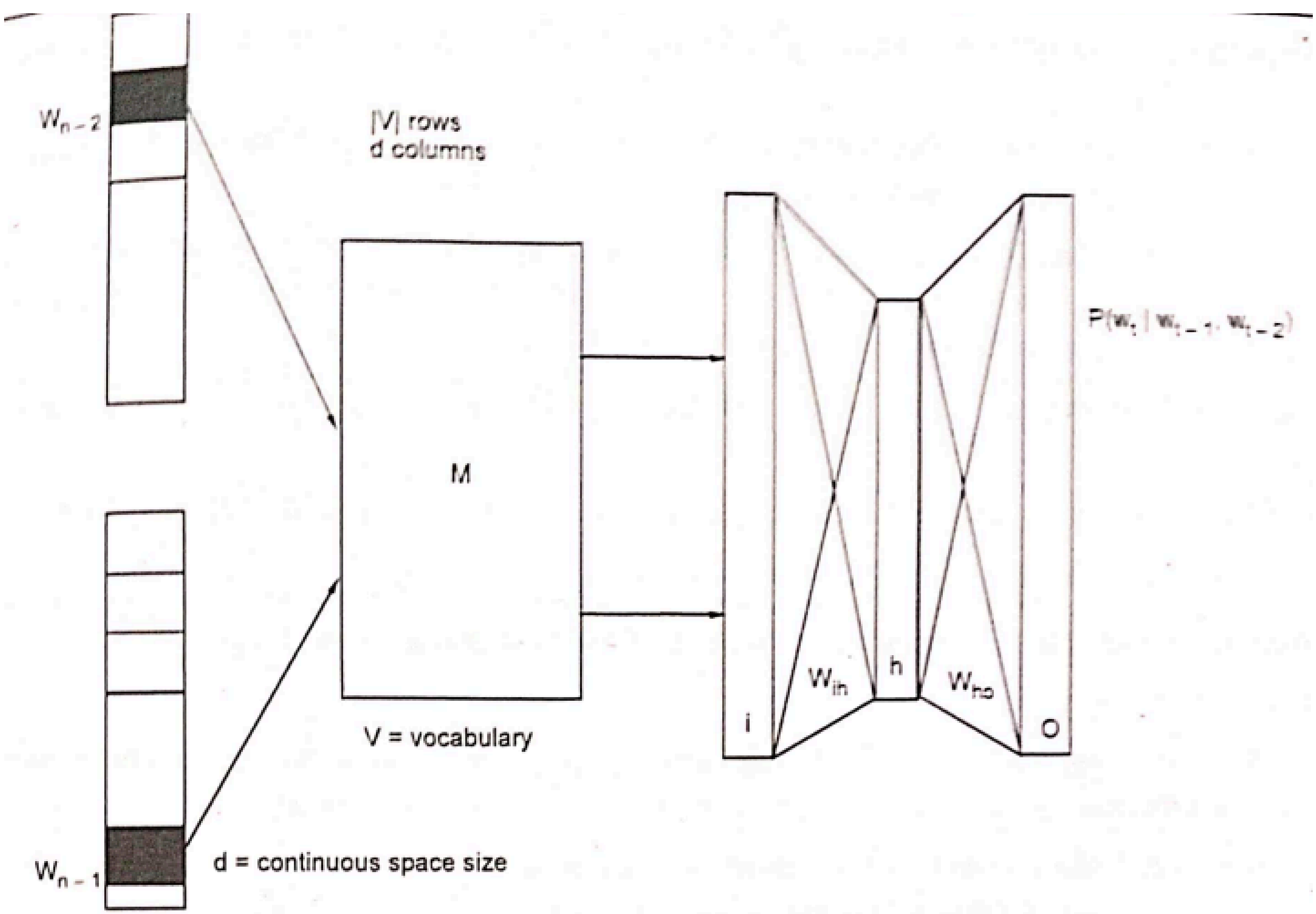

Fig. Q.17.1 : Neural network language model

Formula

$$P(w_t|w_{t-1},...,w_{t-n+1}) = \text{softmax}(Wh_t + b)$$

where:

- h_t is the hidden state.
- W and b are model parameters.

Example Models

✅ **Word2Vec**: Converts words into **vectors**.

✅ **Transformer-based models** (BERT, GPT).

8. Tree-Based Models

These models **cluster words** based on **hierarchical trees**.

A. Hierarchical Class-Based Models

- Similar to **class-based models**, but with **multi-level hierarchy**.

$$P(w) = P(C_1)P(C_2|C_1)P(w|C_2)$$

B. Random Forest Models

- Uses **decision trees** for word prediction.

Example

- Decision tree:
  - "The stock market" → Predicts "**crashes**" or "**rises**" based on past data.

9. Latent Dirichlet Allocation (LDA)

A **topic-based model** where each document is a **mixture of topics**.

Formula

$$P(w|\theta, \beta) = \sum_z P(w|z, \beta) P(z|\theta)$$

where:

- θ = topic distribution per document.
- β = word distribution per topic.
- $P(z|\theta)$ = probability of a topic given a document.

Example

- LDA identifies topics in news articles:
  - **Topic 1:** "government, law, policy"
  - **Topic 2:** "sports, game, match"

Need for Multilingual Language Modeling

Multi-lingual modeling refers to training a single model that can understand, process, and generate text in multiple languages. The goal is to create a unified system that performs well across many languages, often by sharing representations and knowledge between languages.

Multilingual Language Models (MLLMs) are essential for various reasons:

1. Bridging the Language Gap

- Many applications require **cross-lingual communication** (e.g., Google Translate).
- Helps people **access information** in languages they don't understand.

2. Low-Resource Language Support

- Some languages (e.g., **Basque, Amharic**) have **limited training data**.
- A multilingual model can **transfer knowledge** from high-resource languages (e.g., English) to low-resource ones.

3. Cross-Lingual Transfer Learning

- Learning from one language (e.g., **English**) can **benefit** another (e.g., **Hindi**).
- Useful in **sentiment analysis**, **text classification**, and **speech recognition**.

4. Multilingual Search and Retrieval

- Improves **cross-lingual search engines** (e.g., searching in English but retrieving documents in Spanish).

5. Efficiency in Model Deployment

- Instead of **training separate models** for each language, a single **multilingual model** handles all.
- Reduces **computational and storage costs**.

Cross-Language Modeling

Cross-Language Modeling (CLM) is a type of model that learns to **understand and generate content** across **different languages**, often using data from one language to enhance understanding in another. This type of model can be particularly useful in situations where a task needs to be performed across multiple languages without having dedicated models for each language.

Need for Cross-Language Modeling

1. **Scaling to Multiple Languages**:
    - A **single model** can perform tasks like translation, sentiment analysis, and question answering for **multiple languages**, rather than needing to create separate models for each.
2. **Low-Resource Language Support**:
    - Many languages have limited training data (i.e., **low-resource languages**), so leveraging data from **high-resource languages** (e.g., English) can help improve performance for languages with limited data.
3. **Improved Transfer Learning**:
    - CLM enables knowledge transfer from **one language** to **another**, enhancing tasks like **document classification**, **named entity recognition (NER)**, and **machine translation (MT)**.
4. **Language Agnostic Applications**:
    - Cross-language models can facilitate **multilingual applications**, where content might be provided in one language but processed or output in another (e.g., cross-lingual search engines, multilingual chatbots).

Challenges in Cross-Language Modeling

1. **Language Structure Differences**
    - **Syntax, morphology, and grammar** vary significantly across languages, which makes **direct transfer** of learning from one language to another difficult.

2. **Data Scarcity in Low-Resource Languages**
    - While **high-resource languages** like English, Spanish, and Chinese have vast amounts of data, many languages have limited resources, which can hinder model performance.
3. **Alignment Issues**
    - **Word alignment** between languages can be tricky, especially when languages have no direct **one-to-one word mappings** (e.g., languages with different alphabets or sentence structures).
4. **Semantic Gaps**
    - Some languages may have concepts or idioms that are difficult to map to other languages due to **cultural or contextual differences**.

Key Differences Between Multi-lingual and Cross-language Modeling

Aspect	Multi-lingual Modeling	Cross-language Modeling
Scope	Handles multiple languages within a single model.	Focuses on transferring knowledge between languages.
Training Data	Uses data from multiple languages simultaneously.	Often relies on parallel data or transfer from one language to another.
Goal	General-purpose modeling for many languages.	Enabling tasks across languages, especially for low-resource languages.
Example Use Case	A single model for sentiment analysis in 50 languages.	Using an English model to perform NER in Swahili.