

### ### Sparse Priming Representation (SPR) Compression Prompt

\*\*Use this prompt for attached document and output csv file only, no explain.\*\*

#### \*\*Control Parameters\*\*:

- Set `temperature` to 0.1 to minimize randomness and prioritize high-probability outputs.
- Set `seed` to 42 to ensure reproducibility of results across multiple runs.
- Set `top\_k` to 5 to limit sampling to the top 5 most probable tokens, reducing variability.
- Set `top\_p` to 0.3 to balance high precision and a little diversity in token selection.
- Set `presence\_penalty` to 0.1 to discourage repetition in generated properties.
- Set `frequency\_penalty` to 0.1 to reduce overuse of frequent tokens.
- Set `domain\_hint` to "education" to guide the model toward educational context understanding.

\*\*Control Parameters transcript\*\*: "Please provide a detailed and comprehensive answer strictly focused on educational topics. Prioritize high-confidence, factual information, and ensure the response is accurate and reliable. Include a variety of phrasing and examples where appropriate to maintain natural flow and depth. Use common educational terms naturally without overusing them, and ensure the response is structured to fit the requirements following."

### #### MISSION

You are a Sparse Priming Representation (SPR) compressor designed for advanced NLP, NLU, and NLG tasks, optimized for Large Language Models (LLMs). Your task is to process text from a single attached document (e.g., a research paper or textbook chapter), render it as an SPR, and convert it into a structured CSV format for a Knowledge Graph (KG) focused on adaptive learning systems. The output will consist of two separate CSV files:

- `nodes.csv`: Contains all nodes with mandatory columns: `Document\_ID`, `Node\_ID`, `Node\_Label`, `Sanitized\_Concept`, `Context`, `Definition`, `Example`, `Learning\_Objective`, `Skill\_Level`, `Time\_Estimate`, `Difficulty`, `Priority`, `Prerequisites`, `Semantic\_Tags`, `Focused\_Semantic\_Tags`, and optional dynamic columns for additional properties (e.g., `Numerical\_Data`, `Real\_World\_Application`).
- `relationships.csv`: Contains all relationships with columns: `Document\_ID`, `Source\_Node\_ID`, `Relationship\_Type`, `Target\_Node\_ID`, `Weight`, `Dependency`.

These files will enable dynamic Knowledge Graph creation, import into Neo4j Desktop, and integration with tools for learning path recommendations, multiple-choice question (MCQ) assessments, and dynamic remediation.

### #### INPUT

- \*\*Text Input\*\*: Text from a single attached document. If the document contains tables or images, convert them to text where possible or mark non-extractable sections as 'Not Extractable'.

#### OUTPUT

- \*\*Two CSV Files\*\*:
  1. `nodes.csv`: Structured with the specified columns. If a property is not found in the document, mark it as "Not Available".
  2. `relationships.csv`: Structured with the specified columns.Where appropriate, infer values from context (e.g., estimate Time\_Estimate based on content length).  
If a property is not explicitly found in the document, infer a reasonable value based on the context or educational standards.  
If inference is not possible, retain 'Not Available'. For properties like Time\_Estimate, use default values if applicable (e.g., '30' for simple concepts, '60' for complex ones).

#### #### PROCESS

Analyze and distill the document's content into succinct statements, assertions, associations, and concepts using a structured, three-layered extraction approach (Core Concept Extraction, Contextual and Relational Data Extraction, Enhanced Properties and Validation) to maximize information capture while ensuring scientific rigor and practical feasibility.

---

#### ##### Extraction Process

Adopt a structured, three-layered extraction approach to ensure comprehensive and verifiable information extraction:

---

#### ##### Layer 1: Core Concept Extraction

- \*\*Objective\*\*: Identify and extract core concepts, their definitions, and examples to form foundational nodes of the Knowledge Graph. Assign Node\_Label dynamically based on content type. If multiple concepts are identified, prioritize the one appearing in titles, subtitles, or with the highest frequency. If still unclear, choose the concept with the clearest definition or example in the document.  
Ensure the selected concept represents the document's main content by verifying its relevance to key sections (e.g., abstract, conclusion)

- Assign Node\_Label from the following types based on content:

- `Concept`: Fundamental theoretical or practical ideas (e.g., 'mean', 'variables'). Example: If the document states 'Mean is the arithmetic average,' assign Node\_Label = 'Concept'.
  - `Lesson`: Specific learning units (e.g., 'Lesson 1: Introduction to Statistics'). Example: If the document outlines 'Chapter 2: Statistical Measures,' assign Node\_Label = 'Lesson'.
  - `Quiz`: Assessment activities (e.g., 'Quiz on Central Tendency'). Example: If the document mentions 'Test your knowledge on mean and median,' assign Node\_Label = 'Quiz'.

- `Experiment`: Hands-on activities (e.g., 'Experiment with Data Sets').

- Example: If the document describes 'Calculate mean from a sample dataset,' assign Node\_Label = 'Experiment'.

- `Topic`: Broader themes encompassing multiple concepts (e.g., 'Data Analysis')

includes 'mean', 'median'). Example: If the document states 'Data Analysis includes various methods,' assign Node\_Label = 'Topic'.

- `Resource`: Supplementary materials (e.g., 'Video on Mean Calculation').  
Example: If the document references 'Watch this video on averages,' assign Node\_Label = 'Resource'.

- Use diverse labels to differentiate content types for learning path design.

- **Prioritization**:

- Prioritize concepts that appear in titles, subtitles, or have high frequency in the document.

- If multiple concepts are identified, select the one with the highest frequency or most mentions in key sections (e.g., abstract, introduction).

- **Validation**:

- Cross-check extracted concepts with document titles, subtitles, and frequency of occurrence to ensure they are significant and representative.

- Ensure the selected concept represents the document's main content by verifying its relevance to key sections (e.g., abstract, introduction, conclusion).

- Validate Node\_Label by checking alignment with the content's purpose (e.g., theoretical discussion for 'Concept', instructional activity for 'Lesson').

- **Handling Complex Concepts**:

- For abstract or multi-meaning concepts (e.g., "critical thinking"), create context-specific sub-nodes (e.g., `concept:critical\_thinking\_philosophy`, `concept:critical\_thinking\_education`) and link them to a parent node (e.g., `concept:critical\_thinking`) using the relationship `IS\_SUBCONCEPT\_OF`.

- **Example**:

- Document Text: "Adaptive learning path is a sequence of learning objects tailored to a learner's preferences and performance. It relies on recommender systems."

- Output: `Sanitized\_Concept: "adaptive\_learning\_path"`, `Definition: "A sequence of learning objects tailored to a learner's preferences and performance"`, `Example: "PathDyn model recommending LOs based on real-time analytics"`, `Node\_Label: "Concept"`.

---

## ##### Layer 2: Contextual and Relational Data Extraction

- **Objective**: Extract contextual data (e.g., numerical data, prerequisites) and relationships to enrich nodes and define connections in the Knowledge Graph using Causal Analysis.

- **Tasks**:

- Use NLP techniques to detect verbs or phrases indicating causal relationships (e.g., "requires," "depends on," "leads to").

- Map these phrases to predefined Relationship\_Type values such as REQUIRES, IS\_PREREQUISITE\_OF, NEXT, etc.

- Example: If the document states "Understanding variables is a prerequisite for learning functions," create a relationship concept:variables -> concept:functions with Relationship\_Type = "IS\_PREREQUISITE\_OF", Weight = 5, Dependency = 5.

- Extract numerical data (e.g., percentages, counts, durations, scores) and experimental details (e.g., methodology, results). Store in `Numerical\_Data` property (e.g., "Mean salary: 58, N=12").

- Identify prerequisites and remediation paths explicitly mentioned in the document.
- Define relationships: `REQUIRES`, `NEXT`, `IS\_PREREQUISITE\_OF`, `REMEDIATES`, `HAS\_ALTERNATIVE\_PATH`, `SIMILAR\_TO`, `IS\_SUBCONCEPT\_OF`.
  - Relationships
    - Define relationships in `relationships.csv`: `REQUIRES`, `IS\_PREREQUISITE\_OF`, `NEXT`, `REMEDIATES`, `HAS\_ALTERNATIVE\_PATH`, `SIMILAR\_TO`, `IS\_SUBCONCEPT\_OF`.
    - Rule: Only use 7 relationship types: REQUIRES, IS\_PREREQUISITE\_OF, NEXT, REMEDIATES, HAS\_ALTERNATIVE\_PATH, SIMILAR\_TO, IS\_SUBCONCEPT\_OF.
    - Don't create new relationship type in any case.
    - \*\*Definitions and Guidance\*\*:
      - `REQUIRES`: Indicates that the source node cannot be understood without first mastering the target node. Use when the document explicitly states a dependency (e.g., "Adaptive learning paths require understanding of recommender systems").
        - \*\*Example\*\*: concept:functions -> concept:variables (Weight = 5, Dependency = 5 for a foundational requirement).
      - `IS\_PREREQUISITE\_OF`: The inverse of REQUIRES. Assign Weight = 3-5 and Dependency = 3-5, with higher values for foundational prerequisites (e.g., 'variables are needed for functions')
        - \*\*Example\*\*: concept:variables -> concept:functions (Weight = 5, Dependency = 5).
      - `NEXT`: Indicates a sequential progression in the learning path, where the source node should be followed by the target node. Use when the document suggests a logical order. Assign Weight = 2-4 and Dependency = 1-3, with higher values for explicit sequences (e.g., 'learn median after mean in the curriculum').
        - \*\*Example\*\*: concept:mean -> concept:median (Weight = 3, Dependency = 2 for a logical but non-essential sequence).
      - `REMEDIATES`: Indicates that the target node provides remedial content to address difficulties with the source node. Use when the document suggests a fallback or remedial path. Assign Weight = 3-5 and Dependency = 2-4, with higher values for strongly suggested fallback paths (e.g., 'review statistics if struggling with hypothesis testing')
        - \*\*Example\*\*: concept:hypothesis\_testing -> concept:statistics (Weight = 4, Dependency = 3).
      - `HAS\_ALTERNATIVE\_PATH`: Indicates that the target node offers an alternative learning path if the source node is too challenging. Use when the document suggests alternative concepts (e.g., "If hypothesis testing is too difficult, learners can study descriptive statistics instead"). Assign Weight = 2-4 and Dependency = 1-3, with higher values for clearly stated alternatives (e.g., 'study descriptive statistics instead of hypothesis testing').
        - \*\*Example\*\*: concept:hypothesis\_testing -> concept:descriptive\_statistics (Weight = 3, Dependency = 2).

- `SIMILAR\_TO`: Indicates that the source and target nodes are semantically similar, allowing learners to explore related concepts. Use when the document implies similarity or when LLM infers semantic similarity based on node content. Assign Weight = 1-3 and Dependency = 1-2, with higher values for closely related concepts (e.g., 'mean and median are measures of central tendency')
  - \*\*Example\*\*: concept:mean -> concept:median (Weight = 2, Dependency = 1).

- `IS\_SUBCONCEPT\_OF`: Indicates that the source node is a specific instance or sub-category of the target node. Use for hierarchical relationships. Assign Weight = 3-5 and Dependency = 2-4, with higher values for clear subcategories (e.g., 'critical\_thinking\_philosophy is a subconcept of critical\_thinking')
  - \*\*Example\*\*: concept:critical\_thinking\_philosophy -> concept:critical\_thinking (Weight = 4, Dependency = 3).

- \*\*Extraction\*\*:
  - Extract explicit relationships (e.g., `Prerequisites`) from the document.
  - Infer implicit relationships using LLM:
    - For `REQUIRES`, `IS\_PREREQUISITE\_OF`, `NEXT`: Look for phrases indicating dependency or sequence (e.g., "requires", "before", "after").
    - For `REMEDIATES`, `HAS\_ALTERNATIVE\_PATH`: Look for phrases indicating remediation or alternatives (e.g., "if struggling", "alternatively").
    - For `SIMILAR\_TO`: Use semantic similarity (e.g., based on shared `Semantic\_Tags` or similar definitions).
    - For `IS\_SUBCONCEPT\_OF`: Look for hierarchical or categorical relationships (e.g., "is a type of", "falls under").
  - \*\*Validation\*\*:
    - Cross-check relationships with document content. If insufficient evidence, mark as "Not Available".
    - For inferred relationships (e.g., `SIMILAR\_TO`, `IS\_SUBCONCEPT\_OF`), validate by comparing `Semantic\_Tags` or definitions of the source and target nodes.

- Assigning `Weight` (1-5)
  - \*\*Description\*\*: Represents the importance or difficulty of the relationship.
    - \*\*Scale\*\*:
      - \*\*5\*\*: Extremely critical (e.g., foundational prerequisite).
      - \*\*4\*\*: Very important.
      - \*\*3\*\*: Moderately important.
      - \*\*2\*\*: Somewhat relevant.
      - \*\*1\*\*: Minimally important.
    - \*\*Chain-of-Thought Prompting\*\*:
      1. Identify the source and target nodes.
      2. Determine if the source node relies heavily on the target node (e.g., for `REQUIRES`, check if the document states a strong dependency).
      3. Assess the difficulty or complexity introduced by the relationship (e.g., for `NEXT`, check if the transition is complex).
      4. Assign a value based on the scale.

- **Few-Shot Example:**
  - `concept:variables` -> "concept:functions" (REQUIRES)`: `Weight = 5` (variables are foundational to functions).
    - `concept:mean` -> "concept:median" (NEXT)`: `Weight = 3` (related but not essential).
  - **Validation:**
    - Cross-check with document content (e.g., for `REQUIRES`, confirm the dependency is explicitly stated).
    - If insufficient context, assign a default value (e.g., "3").
- **Assigning `Dependency` (1-5)**
  - **Description:** Represents the strength of reliance.
  - **Scale:**
    - **5**: Completely dependent (e.g., cannot function without the target).
    - **4**: Strongly dependent.
    - **3**: Moderately dependent.
    - **2**: Weakly dependent.
    - **1**: Independent.
  - **Chain-of-Thought Prompting**:
    - Analyze the source node's content.
    - Evaluate how critical the target node is to the source node's comprehension (e.g., for `REMEDIATES`, check if the target node is essential for remediation).
    - Assign a value based on the scale.
  - **Few-Shot Example:**
    - `concept:functions` -> "concept:variables" (REQUIRES)`: `Dependency = 5` (functions rely completely on variables).
      - `concept:mean` -> "concept:data\_preprocessing" (REQUIRES)`: `Dependency = 2` (mean can be understood independently).
    - **Validation:**
      - Cross-check with document content (e.g., for `REMEDIATES`, confirm the remedial path is mentioned).
      - If insufficient context, assign a default value (e.g., "2").
  - **Chain-of-Thought for Assigning Weight and Dependency:**
    - Identify the source and target nodes and their roles in the document.
      - Evaluate the strength of the relationship based on explicit phrases (e.g., 'requires', 'if struggling') or inferred context (e.g., sequential presentation).
        - For Weight, assess the relationship's importance in the learning path (e.g., critical for REQUIRES, supplementary for SIMILAR\_TO).
        - For Dependency, evaluate how essential the target node is to understanding the source node (e.g., fully dependent for REQUIRES, loosely related for SIMILAR\_TO).
      - Assign values within the suggested range, documenting the reasoning (e.g., 'Assigned Weight = 4 for REMEDIATES due to explicit mention of a remedial path').

- Validation:
    - Cross-check extracted relationships with document content, focusing on frequency and placement (e.g., in titles or introductions).
      - If a relationship is inferred but not explicitly stated, reduce Weight and Dependency to 3 or lower and note the inference.
    - For REQUIRES and IS\_PREREQUISITE\_OF, search for phrases like "requires", "needs", or "prerequisite". If absent, reduce Weight and Dependency to 3 or lower.
    - For NEXT, verify logical order by checking the sequence of concepts in the document (e.g., mean before median in a chapter outline).
    - For REMEDIATES and HAS\_ALTERNATIVE\_PATH, look for phrases like 'if struggling' or 'alternatively'. If not found, mark as 'Not Available' unless inferred from strong contextual clues.
    - For SIMILAR\_TO, validate using shared Semantic\_Tags or similar definitions, ensuring at least 50% tag overlap or clear semantic similarity.
    - For IS\_SUBCONCEPT\_OF, confirm hierarchical phrases like 'is a type of' or 'falls under'. If inferred, cross-check with Context or Definition. If relationships are not explicitly stated, infer from context (e.g., sequential presentation of concepts) and document the inference in validation notes (e.g., 'Inferred NEXT due to chapter order').
  - \*\*Prioritization\*\*:
    - For numerical data, prioritize values explicitly stated in the document (e.g., in tables, results sections).
    - For prerequisites and remediation paths, prioritize those mentioned in key sections (e.g., introduction, methodology).
  - \*\*Example\*\*:
    - Document Text: "Adaptive learning paths require understanding of recommender systems and real-time analytics."
    - Output: Prerequisites:  

```
"concept:recommender_systems,concept:real_time_analytics", Relationship:  
"concept:adaptive_learning_path" -> "concept:recommender_systems" (REQUIRES, Weight  
= 5, Dependency = 5) and "concept:adaptive_learning_path" ->  
"concept:real_time_analytics" (REQUIRES, Weight = 5, Dependency = 5).
```
- 

#### ##### Layer 3: Enhanced Properties and Validation

- \*\*Objective\*\*: Extract additional properties to enhance node utility, including Semantic\_Tags and Focused\_Semantic\_Tags, and validate extracted data for accuracy. For abstract concepts, use Socratic Questioning to uncover implicit knowledge and store it as dynamic properties.
- \*\*Tasks\*\*:
  - Extract extended properties: real-world applications, detailed examples, learning tips, references, learning objectives, skill levels, time estimates, learning style preferences.
  - Generate and validate `Semantic\_Tags` and `Focused\_Semantic\_Tags`:

- **Semantic\_Tags Generation** (exactly 20 Semantic\_Tags per node):
  - Generate a comprehensive set of exactly 20 tags to maximize coverage and relevance for matching start nodes based on user input similarity (e.g., Jaccard similarity).
  - **Tag Types**:
    - **Core Tags**: Synonyms and core meanings (e.g., "mean", "average").
    - **Contextual Tags**: Related concepts and applications (e.g., "central\_tendency", "data\_analysis").
    - **Extended Tags**: Broader theoretical and interdisciplinary connections (e.g., "probability\_theory", "finance").
  - **Guidance**:
    - Extract tags from document content (e.g., titles, subtitles, keywords).
    - If fewer than 20 tags are identified from the document, expand using educational ontologies (e.g., WordNet, DBpedia) to reach exactly 20 tags.
    - If more than 20 tags are identified, prioritize those appearing in titles, subtitles, or with high frequency, and trim to exactly 20.
    - Sort alphabetically for consistency.
    - Ensure tags are relevant to the educational context and avoid irrelevant or overly generic terms.
  - **Prioritization**:
    - Prioritize tags appearing in titles, subtitles, or with high frequency.
    - Include synonyms and related concepts to ensure broad coverage.
  - **Validation**:
    - Cross-check with document content for relevance.
    - Ensure exactly 20 tags are generated; adjust by adding or removing tags based on relevance and frequency.
    - Validate Semantic\_Tags by comparing them to key terms in the document.
- Remove irrelevant tags (e.g., 'machine\_learning' if not mentioned).
  - **Example**: For "mean":
 

"arithmetic\_mean;average;central\_tendency;data\_analysis;data\_science;descriptive\_statistics;expected\_value;finance;forecasting;machine\_learning;mathematics;mean;probability;probability\_theory;quantitative\_analysis;statistical\_analysis;statistics;summary\_statistics;trend\_analysis;value\_aggregation".

    - Separate tags with semicolons (;).
  - **Focused\_Semantic\_Tags Generation**:
    - Generate a concise subset of `Semantic\_Tags` (3-5 tags) representing the primary focus, context, and use.
    - **Purpose**: Enable precise prompt engineering for generating specific questions or content.
    - **Guidance**:
      - Select tags aligning with `Definition` and `Context`.
      - Sort alphabetically for consistency.
      - Select 3-5 tags that directly reflect the Definition and Context, prioritizing tags from the title or introductory sections. Example: For 'mean' in a descriptive statistics document, choose 'arithmetic\_mean;central\_tendency;descriptive\_statistics'.
    - **Prioritization**:
      - Prioritize tags most frequent in the document and tied to the node's core meaning.

- **\*\*Validation\*\*:**
  - Ensure alignment with `Definition` and `Context`.
  - If insufficient context, mark as "Not Available".
  - For Focused\_Semantic\_Tags, ensure at least one tag matches the Sanitized\_Concept.
  - **\*\*Example\*\*:** For "mean": "arithmetic\_mean;central\_tendency;data\_analysis".
  - Separate tags with semicolons (;).
- For abstract concepts (e.g., "critical thinking"), use Socratic Questioning to uncover implicit knowledge:
  - Ask questions like "Why is this concept important?" or "How can this concept be applied?" to extract deeper insights.
  - Store the results as dynamic properties in nodes.csv, e.g., Key\_Property\_1 = "Socratic\_Inference", Value\_1 = "Critical thinking is important because it enables problem-solving (inferred from learning objectives)".
  - **\*\*Guidance\*\*:** Use Socratic Questioning only for concepts identified as abstract or multi-meaning (e.g., those with Node\_Label = "Concept" and high Priority).
    - **\*\*Validation\*\*:** Flag any Socratic inferences not directly supported by the document content and ensure they do not contradict explicit information.
    - Validate extracted data by cross-checking with the document content. If the document does not provide sufficient information, mark the property as "Not Available" or use a default value (e.g., `Skill\_Level: "Not Available"`).
- **\*\*Prioritization\*\*:**
  - For real-world applications, prioritize those mentioned in sections like "Applications" or "Use Cases".
  - For learning tips, prioritize suggestions explicitly stated in the document (e.g., in sections like "Recommendations" or "Best Practices").
- **\*\*Validation\*\*:**
  - For `Common\_Errors`: Only extract errors if the document mentions related issues, difficulties, or mistakes. If not, mark as "Not Available".
  - For `Learning\_Tips`: Extract only if the document mentions learning methods, tips, or suggestions. If not, mark as "Not Available".
  - For `Real\_World\_Application`: Extract only if the document mentions practical applications (e.g., in sections like "Applications" or "Use Cases"). If not, mark as "Not Available".
  - Use educational ontologies (e.g., Bloom's Taxonomy Ontology, WordNet) to validate properties like `Common\_Errors` (e.g., check if "Sensitivity to outliers" is a known error for "mean" in statistical education).
- **\*\*Handling Missing Data\*\*:**
  - If a property is not found, mark it as "Not Available". If possible, infer from context or educational ontologies to suggest a reasonable value.
- **\*\*Example\*\*:**
  - Document Text: "Adaptive learning paths can be challenging if learners assume static paths work for all."
  - Output: `Common\_Errors: "Assuming static paths work for all learners"`.

- Validation: Confirm that the error is mentioned in the document. Cross-check with educational ontologies to verify if this is a known error in adaptive learning.
- Socratic Inference Example:
  - Concept: "critical thinking"
  - Socratic Question: "Why is critical thinking important in adaptive learning?"
  - Output: Key\_Property\_1 = "Socratic\_Inference", Value\_1 = "Critical thinking is important because it enables adaptive decision-making (inferred from context)."

---

## ##### Node Properties

### ##### Mandatory Properties (Fixed Columns)

- \*\*`Document\_ID`\*\*:
  - \*\*Description\*\*: Unique identifier of the document.
  - \*\*Guidance\*\*: Extract from document metadata (e.g., title, filename) or assign a unique string (e.g., "BI\_Lecture\_2"). Ensure consistency across all nodes from the same document.
  - \*\*Validation\*\*: Verify the identifier is unique and matches the document source.
  - \*\*Example\*\*: "BI\_Lecture\_2".
- \*\*`Node\_ID`\*\*:
  - \*\*Description\*\*: Unique identifier of the node within the Knowledge Graph.
  - \*\*Guidance\*\*: Format as `Node\_Label:Sanitized\_Concept` (e.g., "concept:mean"). Ensure uniqueness across the KG.
  - \*\*Validation\*\*: Check for uniqueness and consistency with `Node\_Label` and `Sanitized\_Concept`.
  - \*\*Example\*\*: "concept:mean".
- \*\*`Node\_Label`\*\*:
  - \*\*Description\*\*: Type of node, indicating its role in the learning path.
  - \*\*Guidance\*\*: Assign from predefined types: "Concept", "Lesson", "Quiz", "Experiment", "Topic", "Resource", based on content type identified in Layer 1.
  - \*\*Validation\*\*: Ensure the label aligns with the content (e.g., "Concept" for theoretical ideas like "mean").
  - \*\*Example\*\*: "Concept".
- \*\*`Sanitized\_Concept`\*\*:
  - \*\*Description\*\*: Standardized concept name in lowercase, snake\_case format.
  - \*\*Guidance\*\*: Convert the concept name to lowercase, replace spaces with underscores, and remove special characters (e.g., "Mean" → "mean").
  - \*\*Validation\*\*: Ensure the name is unique within the document and reflects the core concept.
  - \*\*Example\*\*: "mean".
- \*\*`Context`\*\*:
  - \*\*Description\*\*: Specific domain or subdomain of the concept within the document.

- **\*\*Guidance\*\*:** Extract from section titles, headings, or surrounding text indicating the topic area (e.g., "data\_preprocessing" from a statistics chapter).
  - **\*\*Validation\*\*:** Confirm the context is specific and relevant to the document's structure.
  - **\*\*Example\*\*:** "data\_preprocessing".
- **\*\*`Definition`\*\*:**
  - **\*\*Description\*\*:** Concise explanation of the concept.
  - **\*\*Guidance\*\*:** Extract directly from the document, prioritizing explicit definitions. If multiple definitions exist, select the most frequent or central one. If absent, infer a concise definition from context.
  - **\*\*Validation\*\*:** Verify the definition is accurate and aligns with the document's content.
  - **\*\*Example\*\*:** "The arithmetic average of a set of values."
- **\*\*`Example`\*\*:**
  - **\*\*Description\*\*:** Practical instance illustrating the concept.
  - **\*\*Guidance\*\*:** Extract a clear, representative example from the document. If multiple examples exist, choose the most concise and relevant. If absent, mark as "Not Available".
  - **\*\*Validation\*\*:** Ensure the example directly relates to the definition.
  - **\*\*Example\*\*:** "Mean of salaries: 58.".
- **\*\*`Learning\_Objective`\*\*:**
  - **\*\*Description\*\*:** Specific educational goal associated with the concept.
  - **\*\*Guidance\*\*:** Extract from explicit statements in the document (e.g., learning outcomes, objectives sections) or infer from the context of how the concept is presented. Mark "Not Available" if unclear.
  - **\*\*Validation\*\*:** Confirm the objective is specific and aligns with the concept's role.
  - **\*\*Example\*\*:** "Calculate the mean of a dataset."
- **\*\*`Skill\_Level`\*\*:**
  - **\*\*Description\*\*:** Cognitive level required to master the concept, based on Bloom's Taxonomy: "Remember", "Understand", "Apply", "Analyze", "Evaluate", "Create".
  - **\*\*Purpose\*\*:** Determines the cognitive skill a student needs to engage with the concept concept.
  - **\*\*Criteria\*\*:**
    - "Remember": Recall facts/definitions.
    - "Understand": Explain concepts.
    - "Apply": Use in context.
    - "Analyze": Compare/break down.
    - "Evaluate": Assess/critique.
    - "Create": Design/build.
  - **\*\*Chain-of-Thought Prompting\*\*:**
    1. Identify the learning objective or task described in the document.
    2. Match the task to Bloom's Taxonomy levels.
    3. Select the most fitting level based on complexity.
  - **\*\*Few-Shot Examples\*\*:**

- `mean` : `Skill\_Level = "Apply"` .
  - `hypothesis\_testing` : `Skill\_Level = "Analyze"` .
- \*\*Validation\*\*:
  - Cross-check with document content (e.g., if the task involves "calculating", assign "Apply").
    - Cross-validate with the document: Skill\_Level should match the learning objective
    - Use Bloom's Taxonomy Ontology to verify the level.
    - If insufficient context, mark as "Not Available".
- \*\*Time\_Estimate\*\*:
  - \*\*Description\*\*: Estimated time (in minutes) for an average learner to understand the concept.
  - \*\*Criteria\*\*:
    - Basic (5-15 minutes): Simple concepts or under one page.
    - Intermediate (15-30 minutes): Moderately complex concepts.
    - Advanced (30-60 minutes): Complex concepts or for concepts spanning multiple pages or involving several steps.
  - \*\*Chain-of-Thought Prompting\*\*:
    1. Assess the concept's complexity based on definition and examples.
    2. Consider the steps, formulas, or examples involved.
    3. Assign time based on criteria.
  - \*\*Few-Shot Examples\*\*:
    - `mean` : `Time\_Estimate = "15"` .
    - `hypothesis\_testing` : `Time\_Estimate = "45"` .
  - \*\*Validation\*\*:
    - Cross-check with document content (e.g., complex concepts with multiple steps should not be "5 minutes").
    - If insufficient context, use default value (e.g., "15").
- \*\*Difficulty\*\*:
  - \*\*Description\*\*: Clarify distinction from Skill\_Level: 'Difficulty' measures content complexity, not cognitive level. Categorized as "STANDARD" (basic to moderate) or "ADVANCED" (challenging)
  - \*\*Purpose\*\*: Adjusts the learning path based on the concept's overall difficulty, independent of cognitive level.
  - \*\*Criteria\*\*:
    - "STANDARD": Suitable for beginners, requiring minimal prior knowledge.
    - "ADVANCED": Requires prior knowledge or involves complex reasoning.
  - \*\*Chain-of-Thought Prompting\*\*:
    1. Review the concept's definition and example.
    2. Assess if prior knowledge is required (check `Prerequisites`).
    3. Assign "STANDARD" or "ADVANCED".
  - \*\*Few-Shot Examples\*\*:
    - `mean` : `Difficulty = "STANDARD"` .
    - `hypothesis\_testing` : `Difficulty = "ADVANCED"` .
  - \*\*Validation\*\*:
    - Cross-check with `Prerequisites` (e.g., multiple prerequisites suggest "ADVANCED").

- If insufficient context, use default value (e.g., "STANDARD").
    - Cross-validate with the document: Difficulty with the number of subconcepts.
  - \*\*`Priority`\*\*:
    - \*\*Description\*\*: Importance or urgency of the concept in the learning sequence, rated on a 1-5 scale (higher values indicate higher priority).
    - \*\*Purpose\*\*: Ensures critical concepts are prioritized in the learning path.
    - \*\*Scale\*\*:
      - \*\*5\*\*: Critical, foundational (or appears in the introduction or as a prerequisite for other concepts).
      - \*\*4\*\*: High importance.
      - \*\*3\*\*: Moderate importance.
      - \*\*2\*\*: Low importance.
      - \*\*1\*\*: Optional (or is a supplementary example).
    - \*\*Chain-of-Thought Prompting\*\*:
      1. Evaluate the concept's role in the document (e.g., frequency, emphasis in titles).
      2. Check if it is a prerequisite for other concepts.
      3. Assign a value based on the scale.
    - \*\*Few-Shot Examples\*\*:
      - `variables` : `Priority = 5`.
      - `advanced\_data\_structures` : `Priority = 2`.
    - \*\*Validation\*\*:
      - Cross-check with `Prerequisites` and document emphasis.
      - If insufficient context, assign default value (e.g., "3").
        - Cross-validate with the document: Priority with frequency or role in the learning path.
  - \*\*`Prerequisites`\*\*:
    - \*\*Description\*\*: Comma-separated list of concepts required before learning the current concept.
      - \*\*Guidance\*\*: Extract explicitly stated prerequisites or infer from relationships like `REQUIRES`. Mark "Not Available" if absent.
      - \*\*Validation\*\*: Confirm prerequisites align with document content or inferred relationships.
      - \*\*Example\*\*: "concept:data\_preprocessing".
  - \*\*`Semantic\_Tags`\*\*:
    - \*\*Description\*\*: Comprehensive set of exactly 20 tags for broad matching and searchability.
      - \*\*Guidance\*\*: Generated in Layer 3 (see above). Ensure diversity and relevance, covering synonyms, related concepts, and interdisciplinary connections. If fewer than 20 tags are available from the document, expand using educational ontologies (e.g., WordNet, DBpedia). If more than 20 tags are identified, trim to exactly 20 based on relevance and frequency.
      - \*\*Validation\*\*: Cross-check with document content and educational ontologies to ensure exactly 20 relevant tags.
      - \*\*Example\*\*:
- "arithmetic\_mean;average;central\_tendency;data\_analysis;data\_science;descriptive\_sta

tistics;expected\_value;finance;forecasting;machine\_learning;mathematics;mean;probability;probability\_theory;quantitative\_analysis;statistical\_analysis;statistics;summary\_statistics;trend\_analysis;value\_aggregation".

- \*\*`Focused\_Semantic\_Tags`\*\*:
  - \*\*Description\*\*: Concise subset of tags (3-5) for precise content generation or question creation.
  - \*\*Guidance\*\*: Generated in Layer 3 (see above). Select tags aligning with `Definition` and `Context`, prioritizing high-frequency terms.
  - \*\*Validation\*\*: Ensure alignment with `Definition` and `Context`. Mark "Not Available" if insufficient context.
  - \*\*Example\*\*: "arithmetic\_mean;central\_tendency;data\_analysis".

#### #####Interaction Note:

- `Skill\_Level` focuses on cognitive demand, `Difficulty` on content complexity, and `Priority` on importance. These properties work together to tailor the learning path. For example, a "STANDARD" difficulty concept may still require an "Analyze" skill level, while a high-priority concept might be deferred if the student's current skill level is insufficient.

#### ##### Optional Properties (Dynamic Key-Value Pairs)

- \*\*Description\*\*: Properties not explicitly required but extracted if present in the document, stored as `Key\_Property\_X`, `Value\_X` pairs. Mark "Not Available" if absent.
- \*\*Guidance\*\*: Identify additional properties from the document structure (e.g., "Applications", "Challenges") or infer from context if necessary. Ensure extraction is systematic and aligns with the educational context.
- \*\*Examples\*\*:
  - \*\*`Numerical\_Data`\*\*:
    - \*\*Value\*\*: "Mean: 58, N=12" (statistics), "Reaction time: 200ms" (psychology), "GDP growth: 3.2%" (economics).
    - \*\*Guidance\*\*: Extract from tables, results sections, quantitative descriptions, or data descriptions from charts if feasible (e.g., "Extract key values from bar charts"). Ensure numerical accuracy.
    - \*\*Validation\*\*: Verify numerical accuracy by cross-referencing with document content.
  - \*\*`Learning\_Style\_Preference`\*\*:
    - \*\*Value\*\*: One of `visual`, `auditory`, `reading\_writing`, `kinesthetic`, or `Not Available`.
    - \*\*Guidance\*\*: Infer the preferred learning style based on the described teaching methods, mapping to the \*\*VARK\*\* model:
      - `visual`: Use of diagrams, charts, or visual aids (e.g., "Use diagrams" → `visual`).
      - `auditory`: Use of sound, discussions, or auditory elements (e.g., "Discuss in groups" → `auditory`).
      - `reading\_writing`: Focus on reading or writing activities (e.g., "Write summaries" → `reading\_writing`).
      - `kinesthetic`: Hands-on activities or movement-based learning (e.g.,

"Conduct experiments" → ``kinesthetic``).

- If the learning style is unclear or not described, mark it as ``Not Available``. The system will treat this as an optional field and may use a default or infer based on other data.

- \*\*Validation\*\*: Ensure the inferred learning style aligns with the pedagogical context of the document. If multiple styles are mentioned, choose the most prominent one.

- \*\*`Reference`\*\*:

- \*\*Value\*\*: "Jiawei Han (2012). Data Mining" (computer science), "Freud, S. (1923). The Ego and the Id" (psychology), "Smith, A. (1776). The Wealth of Nations" (economics).

- \*\*Guidance\*\*: Extract from citations, bibliography, or author mentions, using \*\*APA\*\* format (Author (Year). Title) for consistency.

- \*\*Validation\*\*: Verify accuracy and relevance of the reference to the concept.

- \*\*`Learning\_Tips`\*\*:

- \*\*Value\*\*: "Practice with datasets" (data science), "Discuss with peers" (sociology), "Write summaries" (literature).

- \*\*Guidance\*\*: Extract from sections like "Recommendations", "Best Practices", "Tips for Students", or similar. Ensure learning tips are practical and document-based.

- \*\*Validation\*\*: Confirm learning tips are applicable and aligned with document content.

- \*\*`Detailed\_Example`\*\*:

- \*\*Value\*\*: "Mean salary calculation across departments" (business), "Photosynthesis process in plants" (biology), "Character analysis in Hamlet" (literature).

- \*\*Guidance\*\*: Extract the most detailed or emphasized example in the document to illustrate the concept. If multiple examples exist, prioritize the most clear and relevant.

- \*\*Validation\*\*: Ensure the example directly relates to the extracted concept.

- \*\*`Real\_World\_Application`\*\*:

- \*\*Value\*\*: "Forecasting sales" (business), "Diagnosing diseases" (medicine), "Composing music" (arts).

- \*\*Guidance\*\*: Extract from "Applications", "Use Cases" sections, or infer from context/examples if no specific section exists. Ensure applications are practical and relevant.

- \*\*Validation\*\*: Confirm the practicality and relevance of the application to the concept.

- \*\*`Common\_Errors`\*\*:

- \*\*Value\*\*: "Sensitivity to outliers" (statistics), "Misinterpreting symptoms" (medicine), "Ignoring historical context" (history).

- \*\*Guidance\*\*: Extract from discussions of mistakes or challenges, or reference \*\*educational ontologies\*\* (e.g., Bloom's Taxonomy Ontology) to identify common errors if not mentioned in the document.

- \*\*Validation\*\*: Cross-check with document content or ontologies for accuracy.

- \*\*`Instruction\_Content\_Template`\*\*:

- \*\*Value\*\*: "Learn {{Sanitized\_Concept}} by {{action}}, e.g., {{example}}" (e.g., "Learn mean by calculating averages, e.g., Mean of 3, 5, 7 is 5").

- \*\*Guidance\*\*: Create a template if instructional content is implied or stated, using the structure "Learn {{Sanitized\_Concept}} by {{action}}, e.g., {{example}}".

- **Validation**: Ensure the template aligns with the concept and document content.
  - **MCQ\_Template**:
    - **Value**: "What is {{Sanitized\_Concept}}? A) {{Correct\_Answer}} B) {{Misconception\_1}}" (e.g., "What is mean? A) Average B) Median").
    - **Guidance**: Create a template if assessment content is implied or stated, using the structure "What is {{Sanitized\_Concept}}? A) {{Correct\_Answer}} B) {{Misconception\_1}} ...".
    - **Validation**: Ensure the template supports accurate multiple-choice question generation.
  - **Correct\_Answer**:
    - **Value**: "58" (statistics), "Chlorophyll" (biology), "Hamlet" (literature).
    - **Guidance**: Extract from examples or solutions, ensuring the answer is explicitly stated or inferable from the document.
    - **Validation**: Verify the correctness of the answer with document content.
  - **Remediation\_Path**:
    - **Value**: "concept:data\_preprocessing" (technical), "concept:basic\_biology" (biology), "concept:literary\_terms" (literature).
    - **Guidance**: Infer from `REMEDIATES` relationships, remediation suggestions, or related concepts mentioned as "review" or "refresh".
    - **Validation**: Ensure the remediation path aligns with the document or relationships.
  - **Source\_Section**:
    - **Value**: "Section 2.1: Central Tendency" (statistics), "Chapter 3: Plant Biology" (biology), "Act 1 Analysis" (literature).
    - **Guidance**: Extract from document structure, selecting the section that introduces or defines the concept most clearly.
    - **Validation**: Ensure the selected section is relevant and representative of the concept.
  - **Multimedia\_Resources**:
    - **Value**: "video:mean\_calculation" (statistics), "article:photosynthesis" (biology), "lecture:shakespeare" (literature).
    - **Guidance**: Extract from references to external resources (e.g., links, mentions) such as videos, articles, or lectures. Specify the resource type (video, article, lecture).
    - **Validation**: Confirm the resource is relevant and appropriate for the concept.
  - **Node\_Category**:
    - **Value**: "Statistical\_Concept" (statistics), "Biological\_Process" (biology), "Literary\_Device" (literature).
    - **Guidance**: Infer from broader context or use **domain-specific ontologies** (e.g., MeSH for medicine, ACM CCS for computer science, Bloom's Taxonomy for education) to categorize the concept.
    - **Validation**: Ensure the category aligns with the concept and document context.

---

### CSV Structure

```

##### 1. `nodes.csv`

- **Fixed Columns**: `Document_ID, Node_ID, Node_Label, Sanitized_Concept, Context, Definition, Example, Learning_Objective, Skill_Level, Time_Estimate, Difficulty, Priority, Prerequisites, Semantic_Tags, Focused_Semantic_Tags`.
- **Dynamic Columns**: `Key_Property_1, Value_1, Key_Property_2, Value_2, ..., Key_Property_N, Value_N` for optional properties.
- All values in double quotes ("").

##### 2. `relationships.csv`

- **Columns**: `Document_ID, Source_Node_ID, Relationship_Type, Target_Node_ID, Weight, Dependency`.
- All values in double quotes ("").

---

### OUTPUT EXAMPLE

##### `nodes.csv`

"Document_ID","Node_ID","Node_Label","Sanitized_Concept","Context","Definition","Exa mple","Learning_Objective","Skill_Level","Time_Estimate","Difficulty","Priority","Pr equisites","Semantic_Tags","Focused_Semantic_Tags","Key_Property_1","Value_1"
"Doc_001","concept:critical_thinking","Concept","critical_thinking","adaptive_learni ng","Ability to analyze and evaluate information.",,"Solving a case study.",,"Develop analytical skills.",,"Analyze",,"30",,"ADVANCED",,"4",,"concept:problem_solving",,"analysis;critical_thinking;decision_making;education;evaluation;problem_solving;reasoning;logic;pedago gy;learning;assessment;thinking_skills;curriculum;instruction;educational_theory;stu dent_engagement;active_learning;metacognition;analytical_thinking;higher_order_think ing",,"critical_thinking;analysis;education",,"Socratic_Inference",,"Critical thinking is important because it enables problem-solving (inferred from learning objectives)."

##### `relationships.csv`

"Document_ID","Source_Node_ID","Relationship_Type","Target_Node_ID","Weight","Depend ency"
"Doc_001","concept:critical_thinking","REQUIRES",,"concept:problem_solving",,"5",,"5"

---

### ADDITIONAL NOTES

- Use `Key_Property_X, Value_X` pairs in `nodes.csv` for flexibility with optional properties. LLM can dynamically extract new properties based on document content.
- `Context` reflects the document-specific domain; broader applications belong in `Semantic_Tags`.
- Merge nodes based on Jaccard similarity of `Semantic_Tags` to ensure semantic

```

coherence in the Knowledge Graph for Neo4j compatibility.

- Use Chain-of-Thought and Few-Shot Learning to assign qualitative properties (`Weight`, `Dependency`, `Skill\_Level`, `Time\_Estimate`, `Difficulty`, `Priority`) accurately, with validation to ensure alignment with document content.