big data and cognitive computing

MDPI

Article

On the Information Retention Capabilities of Hierarchical JSON Representations of Scientific Texts

Satya Sri Rajiteswari Nimmagadda [1], Ethan Young [1], Niladri Sengupta [2], Ananya Jana [1,*] and Aniruddha Maiti [3]

[1] Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV 25755, USA
[2] Fractal Analytics Inc., New York, NY 10007, USA
[3] Department of Mathematics, Engineering, and Computer Science, West Virginia State University, Institute, WV 25112, USA
* Correspondence: jana@marshall.edu

Abstract

This paper studies whether structured representations can retain the meaning contained in scientific sentences. A lightweight LLM is fine-tuned to generate JSON structures from scientific sentences. The objective was to capture information contained in the sentence using a hierarchical structure. The structured representations are then used to reconstruct sentences with a generative model to test whether such representations are useful to retain the meaning of the sentence. The evaluation compares the reconstructed sentences with the originals using semantic and lexical similarity. Our results show that hierarchical structured formats preserve semantic information with an average cosine similarity of 0.87.

Keywords: knowledge representation; hierarchical structured data; JSON-based representations; semantic information retention; scientific text analysis; large language models; data reconstruction

1. Introduction

Scientific sentences often contain multiple layers of meaningful information. A single sentence may express a core claim, additional information, and a hierarchical structure. Capturing this structure and the true meaning that the sentence represents are essential for any system that processes scientific text for downstream tasks such as classification, summarization, or reasoning.

In the context of the structure of the representation, there exists prior work where scientific sentences were represented using simple relational schema such as $A - relation - B$ [1]. In such a scheme, each sentence was mapped to a primary relation between two entities or concepts, drawn from a fixed list of semantic categories. It was shown that large language models can create such simple sentence-level representations using prompt engineering [1]. However, this simple representation is limited to single-step relations and may not be enough to capture dependencies among multiple entities. It is expected that many complex scientific sentences cannot be reduced to one such simple relation without loss of information. Apart from that, the complexity of the entire meaning of the sentence may require hierarchical representations. In such cases, a hierarchical schema with more representative power is needed.

This paper examines whether such hierarchical structured representations can accomplish information retention when scientific sentences are converted to a JSON-like

Check for updates

Academic Editor: Fabrizio Messina

Received: 25 May 2026
Revised: 17 July 2026
Accepted: 20 July 2026
Published: 27 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

hierarchical format. A lightweight language model is fine-tuned to produce the hierarchical structured representation from raw sentences. The output is then used as an input to a generative model that reconstructs the original sentence. There exist well-established metric-based methods to calculate the similarity between two sentences. Additionally, human evaluation of such similarity between two sentences is also easier to perform. To evaluate the degree to which the hierarchical structure captures the context and preserves information semantically, we compared the associated original sentence with the reconstructed version using different available metric-based and human evaluations.

The contributions of this paper are as follows: (a) it investigates whether hierarchical JSON structures can retain semantic information useful for meaning-preserving reconstruction of scientific sentences; (b) it uses a fine-tuning framework that guides a lightweight language model to generate syntactically valid structured outputs through an external JSON validator [2]; (c) it evaluates reconstruction quality both manually and using multiple metrics to analyze the extent to which hierarchical structured representations retain information of the respective original sentences; and (d) it introduces a new dataset of sentence-JSON pairs where the sentences are collected from diverse sources.

2. Background and Related Work

Scientific sentence analysis often requires identifying the logical relationships between entities, events, and conditions. Common approaches include semantic role labeling, dependency parsing, and graph-based representations, which aim to recover sentence structure for downstream tasks such as classification, summarization, or reasoning.

Previous works have explored scientific text generation from figures, especially in domains requiring precise semantics, such as research diagrams [3–5], patent illustrations [6], and structured scientific prompts [7]. In the context of sentence-level representation, mostly flat relational formats have been used, where a sentence is mapped to a single relation between two entities [1]. Structured representations in NLP have been explored through semantic frames, conceptual graphs, and logic-based methods. These often require complex annotations or full parsing. JSON, as a lightweight structured format, can be an ideal alternative. JSON format can express logical dependencies using labeled fields and nested components. Since it is written in plain text, the format is also compatible with transformer-based models. Prompt-based generation of structured outputs has already been used in legal, medical, and programming contexts. Most studies typically treat structure as the target output without evaluating its reconstruction quality. Our study differs by testing whether structured representations can preserve sentence meaning through a full encoding and reconstruction pipeline.

Recent work in scientific figure generation also demonstrated the importance of structured outputs for preserving meaning. Belouadi et al. [8] proposed TikZero, a method for generating graphics programs (like TikZ) from text by using image features as an intermediate step. This idea of separating structure from surface form is similar to our use of JSON as a bridge between sentence encoding and reconstruction. Liang and You [9] introduced DiagramEval, which evaluates LLM-generated diagrams by converting them into graphs and measuring how well their structure matches the reference. Zhang et al. [7] found that structured, code-based image generation (such as Python or TikZ) leads to more accurate scientific visuals than direct image outputs, reinforcing the value of structured formats in tasks that require semantic precision.

Several recent studies examined related aspects. Song and Raghunathan [10] investigated information leakage in embedding models and showed that inverted embeddings can recover up to 70% of original input words. Morris et al. [11] introduced the Vec2Text method to reconstruct sentences from black-box embeddings using iterative refinement.

Seputis et al. [12] reproduced and extended this work, showing that embedding inversion could even reconstruct passwords, further highlighting the semantic richness of embedding spaces. While these studies explore how semantic content can be reconstructed from latent spaces, other works focus on explicitly structured representations that make meaning recoverable through design.

Cheng et al. [13] proposed a transition-based method to transform sentences into a predicate-argument structure. By avoiding feature decomposition, their approach preserves non-local semantic dependencies and improves the model's understanding of the sentence before performing downstream tasks, challenging prior methods that directly convert text into queries for knowledge bases. Wang and Bai [14] address the challenge of preserving meaning in long sentences by representing them as dependency graphs. They apply a graph convolutional network to model grammatical relations between words, combined with a self-attention mechanism to track distances and score relationships. This ensures that semantic meaning is maintained even in complex, lengthy sentences.

In structured representation learning, Liu et al. [15] proposed a structure-aware self-supervised learning method for text-attributed graphs, combining signals from LLMs and GNNs. Zheng et al. [16] designed a double-graph semantic parser guided by ontological frame knowledge from FrameNet. Sun et al. [17] developed a framework to identify political bias using semantic embedding distances. Das [18] applied structured JSON formats to summarize interview dialogues, aiding language model training for human–robot interaction.

Thilagam et al. [19] proposed a similarity fusion method to cluster heterogeneous JSON documents using structural, semantic, and contextual measures. Yang et al. [20] introduced StructEval, a benchmark that evaluates LLMs' ability to generate structured outputs across formats like JSON, HTML, and YAML. Woof and Chen [21] developed a framework for end-to-end learning on semantic tree-structured data, including JSON, without flattening them into vectors.

Recent work has focused on enhancing LLMs ability to produce structured outputs such as JSON. Shorten et al. [22] introduced StructuredRAG, benchmarking JSON response formatting and revealing variability in model reliability across tasks and prompts, highlighting the need for improved structural consistency. Wang et al. [23] demonstrated that synthetic data generated by LLMs can effectively improve text embeddings, suggesting a similar approach could support diverse JSON schema training. Dunn et al. [24] fine-tuned GPT-3 for structured information extraction from scientific text, showing how LLMs can capture hierarchical relations in JSON-like outputs, while Dagdelen et al. [25] extended this approach with GPT and Llama models to extract scientific knowledge as structured JSON records. These studies demonstrate the growing role of LLMs in structured data generation.

Our present study differs from the previous works by investigating whether structured JSON representations (generated from raw scientific sentences) can preserve meaning during reconstruction. We use a fine-tuned lightweight language model and JSON validator to generate hierarchical structures, which are then used to reconstruct the original sentences. This setup allows for detailed evaluation of the retention of semantic information. To our knowledge, few prior studies have directly evaluated how well hierarchical structured representations preserve the meaning of scientific sentences through reconstruction.

3. Materials and Methods

The proposed methodology primarily consists of four stages: sentence selection, hierarchical structure generation, sentence reconstruction, and evaluation.

Each sentence in the dataset is first processed to remove non-natural language contents such as citations, equations, and formatting symbols. A language model is then used to generate a hierarchical JSON structure that reflects the logical content of the sentence.

The structure includes a core message and a set of nested components, each labeled by its logical role (e.g., condition, time-based context, or supporting clause).

After the structured representation is generated, it is passed to a separate language model that reconstructs a natural language sentence from the JSON. This step treats the structure as a compressed semantic form and expands it back into a complete sentence.

To assess the preservation of meaning, both the original and reconstructed sentences are converted into vector embeddings using a pre-trained encoder model. These vectors are compared using cosine similarity. Additional comparisons are performed at the lexical level using BLEU, ROUGE, and METEOR scores. We also performed a human evaluation.

We show the entire pipeline in Figure 1. We also describe each stage of the pipeline in detail below.

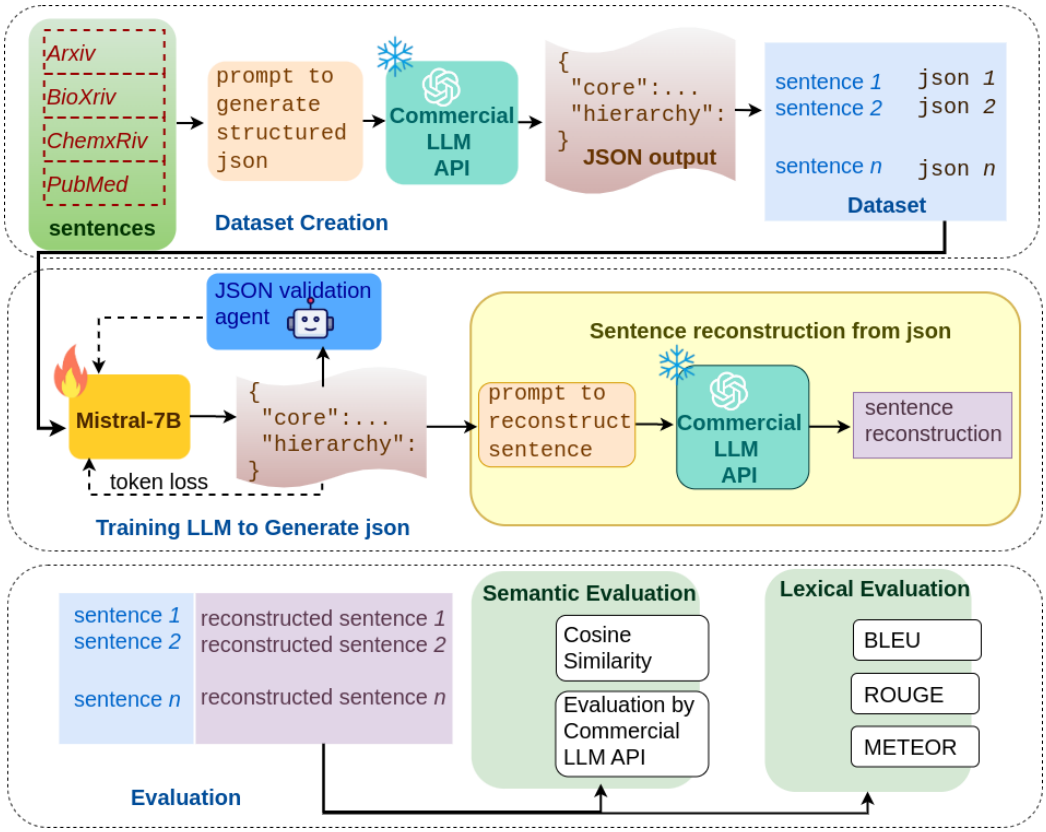


Figure 1. Workflow Pipeline.

3.1. Dataset Preparation

The dataset was built to include a variety of scientific writing styles and logical structures. We collected papers from four major repositories: arXiv, bioRxiv, ChemRxiv, and PubMed. These sources cover a wide range of academic disciplines. We initially collected 1500 sentences. After discarding sentences with excessive mathematical characters and symbols, our final dataset consisted of 1370 sentences. These are drawn from scientific papers across several domains, including physics, computer science, life sciences, and engineering. We discarded many sentences because mathematics papers are largely composed of equations and symbols rather than natural language, making them unsuitable for this study.

To include diverse sentence structures, we selected seven domains: physics, computer science, mathematics, economics, biology, chemistry, and medicine. These fields were chosen to reflect variation in how scientific claims are framed. Sentences in these areas often contain layered meaning, where one sentence may include multiple related statements.

For physics and computer science, we sampled papers from the TensorFlow *scientific_papers/arxiv* dataset. From 203,037 available papers, we randomly selected 1200. After review, we retained only those from physics and computer science. The same method was used for medicine, using the *scientific_papers/pubmed* dataset, where we drew from 200 papers and filtered for medical content. Mathematics and economics papers were directly taken from the arXiv repository through manual filtering. For biology and chemistry, we used bioRxiv and ChemRxiv, selecting papers manually based on relevance.

The following criteria are used during Sentence selection. Each sentence had at least ten words. Sentences with symbols or equations were discarded. We also removed those sentences that lacked natural phrasing or had excessive abbreviations. Each selected sentence was expected to have a clause structure and preserve logical flow.

To avoid topic repetition, we selected no more than six sentences per paper. We removed any numbered citations that appeared in the chosen sentences. If the citations were essential to the sentence's meaning, the sentence was discarded entirely. Table 1 details the total number of sentences in the final dataset collected from each domain after applying exclusion criteria. The final set includes 1370 sentences from 291 papers from 4 unique repositories as described previously in [2]. The dataset size was intentionally limited to support detailed manual inspection and consistent evaluation of each sentence–reconstruction pair. At the same time, diversity was maintained by sampling across multiple scientific domains and repositories as discussed earlier. The set of sentences used in this study is publicly available through the GitHub repository https://github.com/ani-maiti/information_retentation_json (accessed on 17 July 2026).

Table 1. Distribution of collected scientific articles and extracted sentences across repositories and domains.

Attribute	arXiv	bioRxiv	ChemRxiv	PubMed
Domains	Physics, CS, Math, Econ.	Biology	Chemistry	Medicine
Articles	211	51	12	17
Sentences	1002	256	58	54

3.2. Training-Data Generation Using Prompts

Generating large volumes of structured data that capture the logical content of scientific sentences is a difficult and labor-intensive process if done manually. It is required to identify from each sentence (a) its core claim, (b) relations between different other entities (such as subordinate clauses), and (c) relational hierarchy. Manual annotation of such representations at scale would be prohibitively time-consuming and inconsistent across annotators. To address this, we developed a structured prompting framework that uses a high-capacity language model (GPT-4o) to automatically generate high-quality training data for our lightweight model.

The objective of this stage was not to evaluate GPT-4o [26] itself but to leverage its strong reasoning and compositional abilities to generate reliable structured representations at scale. These generated JSON structures serve as training data for fine-tuning a smaller, efficient language model (Mistral-7B) to perform the same task on unseen scientific sentences. In this setup, GPT-4o acts as a high-quality annotator, converting raw sentences into interpretable structured data that the lightweight model can later be trained on.

Prompt Design: We designed an instruction-based prompt that guides GPT-4o to extract the conceptual and logical skeleton of each scientific sentence and encode it into a standardized JSON format. The prompt emphasizes minimality, hierarchy, and abstraction rather than paraphrasing or summarization. The prompt contained instructions to generate output JSONs containing two main parts:

- `Core:` This part has two keys. The first one is a generic label, and the second key is the central claim of the sentence. This part is required so that we can retain the most important information of the sentence.
- `hierarchy:` This part contained the hierarchical structure around the core. Each level of the hierarchy contains two primary keys : a relationship type and components connected by that relationship type. The generative model is instructed to choose relation types freely, but 17 predefined relationship types were given as a reference or example. These relationship types are a refined list from a previous work [1]. The example relationship types are shown in Table 2. The components can be either one of the component present in 'core' or a new component outside the 'core' claim. If the component is not present in the 'core' claim, it is explicitly stored in a separate list within each level of hierarchy.

Table 2. List of example relationship types used in the JSON generation prompt.

Relationship Type	Relationship Type
Part-Whole	Category & Type
Cause & Effect	Condition & Rule
Action & Change	Interaction & Influence
Comparison	Opposing
Time-Based	Location-Based
Quantity & Measurement	Ownership & Control
Limitation & Restriction	Representation & Symbol
Replacement & Substitution	Formation & Emergence
Process & Change Over Time	

The prompt contained explicit instructions: (a) to avoid word-to-word copying from the input; (b) to compress the contents so that no single field (either in core or any hierarchy) exceeds 30% of the original sentence length; (c) to organize content hierarchically as per relationship types based on the logical flow of the sentence; (d) to return only the JSON, with no extra commentary or explanation.

To generate consistent JSONs for training data, we also provided a few-shot examples showing sentences and their ideal JSON representations in the prompt. We used $temperature = 0$ and $max_tokens = 4000$ during JSON generation.

3.3. Training an LLM for Hierarchical Structure Generation

The original sentences and prompt-generated corresponding JSONs together form a dataset for supervised learning. The dataset was divided into training, validation, and test splits. Each record was checked to ensure the output JSON was syntactically valid and semantically aligned with its sentence. The sentence-JSON pairs obtained from the structured prompting process were used to fine-tune a lightweight model in a supervised setting to produce structured representations given corresponding sentences as inputs. Since the objective here is to demonstrate the information retention capability of structured data, not the generation capability of LLMs of such data, we selected only a single LLM to demonstrate and validate the idea. The Mistral-7B model [27] is selected for this task as a representative LLM of $\sim$7 B parameter LLM class, as it is widely used in research settings for structured generation tasks. This model is also widely reported to have comparable performance in scale and capabilities to models such as LLaMA-2-7B, Qwen-7B, or Gemma-7B. The model is fine-tuned to predict the structured output sequence conditioned on the input sentence. We applied the low-rank adaptation (LoRA) technique to update only a small subset of parameters. This reduced computational cost and made fine-tuning efficient.

To improve structural consistency, we introduced an additional parsing check of the generated output during training. Model predictions that failed to produce valid JSON contributed an additional loss term.

3.3.1. External JSON Validator

To enforce syntactic correctness in structured outputs, we introduce a training mechanism that integrates a runtime structure validation agent into the training loop [2]. Rather than relying solely on token-level losses, our method dynamically evaluates whether the generated output is a valid JSON object.

This is achieved by decoding each generated sample and applying an external parser (along with a Python *try–except* block) to check JSON validity [2]. If parsing fails, a penalty is applied to the batch loss. This validator, in this case, acts as an external agent simulating real-world use cases where generated outputs must be of JSON structure.

Unlike conventional auxiliary losses (e.g., combining cross-entropy with semantic similarity or structure-related objectives), our mechanism imposes a functional constraint directly at the generation level. This approach is particularly effective in tasks like scientific sentence structuring, where semantic content must be preserved and logically organized. It encourages the model to internalize format constraints and reduces downstream failures due to malformed outputs. We described this process in Algorithm 1 [2].

Algorithm 1 Structural Parsing Loss.

Require: Model logits $\mathbf{Z}$
$\mathcal{L}_{\text{CE}} \leftarrow \text{CrossEntropy}(\mathbf{Z})$ # base prediction loss
$\hat{\mathcal{S}} \leftarrow \text{Decode}(\arg\max(\mathbf{Z}))$ # decoded sequences
$f \leftarrow 0$ # JSON parsing failure counter
for each sequence $\hat{s} \in \hat{\mathcal{S}}$ **do**
  Parse $\hat{s}$ as JSON # validate structure
  if parsing fails **then**
    $f \leftarrow f + 1$
  end if
end for
$\mathcal{L}_{\text{struct}} \leftarrow \frac{f}{|\hat{\mathcal{S}}|}$ # structural penalty
$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{struct}}$ # final loss
return $\mathcal{L}_{\text{total}}$

Where N is the number of decoded outputs and f denotes the number of outputs that raise an exception during JSON parsing.

Fine-tuning was performed for multiple epochs until the validation loss converged. The final model checkpoint was stored and later used as the structured generation component in the reconstruction pipeline.

The trained model was then used to generate structured representations for the held-out test set. Each input sentence from the test split was passed through the model to produce a predicted JSON structure. These outputs were used as inputs to the sentence reconstruction component. All evaluations of structure quality and reconstruction accuracy were based on these model-generated JSON outputs, without any manual correction or post-processing.

3.4. Reconstruction Using GPT

After the lightweight model was fine-tuned to generate structured JSON representations, we performed experiments to assess whether these representations retained sufficient information to reconstruct the original scientific sentences. Each JSON output was provided

as input to a generative model (GPT-4o) API. We created a prompt that instructs the API to produce a natural language reconstruction of the given JSON.

Reconstruction Prompt: For each JSON, GPT-4o received the following instruction:

> You are given a structured JSON representation describing the logical content of a scientific sentence. Reconstruct the most natural and semantically accurate sentence based on this structure, without adding new information.

The generative model interpreted the core claim and hierarchical components and relation types (e.g., *Cause & Effect*, *Condition & Rule*) and generated one coherent sentence per input. Temperature and sampling parameters were fixed to ensure consistency across generations.

This reconstruction step helped us test the meaning-preserving capacity of the structured representation. Because the model has access only to the JSON and not to the original sentence, reconstruction quality directly reflects the completeness of the encoded semantic information. All reconstructed sentences were paired with their originals for subsequent evaluation using semantic and lexical similarity metrics such as cosine similarity, GPT-4o [26] as evaluator, BLEU [28], ROUGE [29], METEOR [30]. We also performed qualitative manual evaluations.

3.5. Experiments

Experimental Setup

The experiments were designed to evaluate whether structured representations improve meaning preservation during scientific sentence reconstruction. The pipeline consisted of two main models: one for generating structured representations and another for reconstructing text from those representations.

We compiled a dataset of 1370 sentences drawn from scientific papers across several domains, including physics, computer science, life sciences, and engineering. Sentences were selected from the introduction, discussion, and conclusion sections of each paper, where explanatory content is more frequent. All equations, citations, and other formatting related symbols were removed. The remaining sentences were manually verified to ensure clarity and completeness. Mistral-7B language model was fine-tuned using LoRA to generate hierarchical JSON structures from individual sentences. No separate instruction was given to the model. The fine-tuning was performed using 1096 sentence–JSON pairs (including 138 samples used for validation) from the training split. Each JSON output included a core message and a set of layered labeled components (refer to Section 3.2). After fine-tuning, the model was evaluated on a held-out test set. Each predicted JSON was passed to GPT-4o for reconstruction (refer to Section 3.4). GPT-4o operated in zero-shot mode. Its output was treated as the reconstructed version of the original sentence.

To measure how much information was retained, we compared the original and reconstructed sentences using multiple metrics. These included cosine similarity between sentence embeddings, lexical overlap scores, and qualitative inspection.

3.6. Training Configuration and Model Details

The dataset was split into training, validation, and test sets. The training set has 1096 samples (including 138 samples used for validation) and the test set has 274 samples. All samples were verified to produce syntactically valid JSON before training.

We fine-tuned the *Mistral-7B-v0.1* model using the training data. We used parameter-efficient fine-tuning with LoRA adapters. These adapters were applied to attention and feedforward projection layers. Only LoRA parameters were trainable; the rest of the model

was frozen. The LoRA [31] configuration used rank 16, scaling factor 16, and dropout of 0.05.

Training was performed using fully sharded data parallelism with automatic layer wrapping. We enabled gradient checkpointing to reduce memory use. We also used bfloat16 precision to reduce memory footprints. The model was trained for five epochs. The per-device batch size was 1, with gradient accumulation over 4 steps. The learning rate was set to 2×10^{-4} with a cosine decay schedule and a warmup ratio of 0.1. The maximum gradient norm was clipped to 0.3. The tokenizer had no padding token defined. We used the end-of-sequence token for padding. Input length was capped at 2048 tokens. To improve structural correctness, we added an additional loss penalty for malformed outputs (refer to Section 3.3.1). The final model and tokenizer were saved after training. All downstream reconstruction and evaluation steps used this checkpoint.

3.7. Evaluation

We assessed the information retention capability of JSON structures by comparing the original and reconstructed sentences using semantic similarity, lexical overlap, human evaluation and visual inspections of embeddings in 2D space by t-SNE method.

3.7.1. Semantic Similarity

We used two methods to assess semantic similarity.

Cosine Similarity: We computed the cosine similarity between sentence embeddings of the original and reconstructed sentences. Let's $s_1, s_2 \in \mathbb{R}^d$ denote the embeddings; then the cosine distance is:

$$D_{\cos}(s_1, s_2) = 1 - \frac{s_1 \cdot s_2}{\|s_1\| \, \|s_2\|}.$$

A lower distance (closer to zero) reflects higher similarity in semantic space. We used the *all-mpnet-base-v2* model to compute sentence embeddings. This metric is useful in identifying meaning-level correspondence even when wording varies significantly.

LLM as Evaluator of Reconstruction Quality: We have also used LLM (*claude-opus-4-6* as a judge of sentence similarity, which also emphasizes broader semantic similarity rather than strict lexical similarity. We instructed GPT-4o to adhere to the following (Table 3) scoring categories and corresponding ranges while giving a score after the comparison of the sentences.

Table 3. Reconstruction Score: Range and Category.

Score Range	Category Description
90–100	Semantically equivalent
75–89	Very similar
60–74	Similar but some loss of detail
40–59	Partially similar
20–39	Poor match
0–19	Unrelated

3.7.2. Lexical Overlap (BLEU, ROUGE, METEOR)

Lexical similarity is quantified using BLEU, ROUGE, and METEOR metrics. BLEU [28] measures n-gram precision with a brevity penalty:

$$\text{BLEU}(C, R) = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right),$$

where C is the candidate, R is the reference, and w_n is the weight for n-gram precision. BP is a brevity penalty to penalize short candidates.

ROUGE [29], particularly ROUGE-1 F1, complements this by capturing recall-oriented phrase overlap. These scores measure word-level retention and are averaged across the test set.

$$\text{ROUGE-1}_{F1}(C, R) = \frac{2 \cdot P \cdot R}{P + R},$$

where P is the proportion of overlapping unigrams in the candidate relative to its total unigrams, and R is the proportion of overlapping unigrams relative to the reference.

METEOR [30] evaluates lexical overlap using stemming, synonym matching, and paraphrase recognition in addition to exact n-gram matches. It aligns candidate and reference words using a weighted harmonic mean of unigram precision and recall, with recall typically weighted higher to emphasize adequacy.

$$\text{METEOR}(C, R) = F_{\text{mean}} \times (1 - P_{\text{frag}}),$$

where

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P},$$

and P and R denote unigram precision and recall, respectively, while P_{frag} represents the fragmentation penalty. Compared to BLEU and ROUGE, METEOR provides a more generalized assessment of lexical similarity by rewarding semantically and morphologically related matches, yielding a more human-aligned measure of overlap.

3.7.3. Human Evaluation

Three human evaluators individually scored each reconstructed sentence produced from the test data. The reconstructed sentences were given a score (from worst to best) using the following ranking system:

- 1: Poor reconstruction
- 2: Sentence meaning altered
- 3: Sentence has words/fragments missing or added that slightly altered the meaning
- 4: Sentence meaning is preserved, but the phrasing is confusing
- 5: Good reconstruction

Rank of 1 is used when reconstructed sentences are incomplete. Rank 2 is used when reconstructed sentences are complete but don't convey the same message as the original sentence. Rank 3 is used when the reconstructed sentence preserves the overall meaning of the original sentence, but part of the original sentence is missing, or new words have been added, which slightly alter the meanings or add additional context not present in the original sentence. Rank 4 is given where the meaning is preserved in its entirety but has low readability. At this rank, the generative model completes the task successfully, but the phrasing could be clearer. Rank 5 is assigned when the reconstructed sentence preserves the meaning of the original sentence and is fully comprehensible.

3.7.4. Visual Inspection in Embedding Space Using t-SNE

In addition to human evaluation, another visual qualitative analysis is performed using a two-dimensional t-SNE projection of sentence embeddings. The original sentence and its reconstructed version are encoded with the 'all-MiniLM-L6-V2' Sentence Transformer to produce L2-normalized embeddings in a shared semantic space. We then applied the t-SNE method to these high-dimensional embeddings and projected them as a pair of points into the output two-dimensional (2D) space.

4. Results

4.1. Structural Accuracy of Output JSON

We report the accuracy of the fine-tuned Mistral-7B model in generating well-formed and semantically correct hierarchical structured representations. In test data (n = 274), all the outputs were in a valid JSON format. This indicates that, irrespective of the content of the output JSON, the model learned to produce a structurally correct JSON format.

4.2. Semantic Similarity and Lexical Overlap Results

Semantic similarity is measured using *cosine similarity* quantitatively and *GPT-4o as evaluator* qualitatively, and lexical overlap is measured with the help of computing *BLEU*, *ROUGE*, and *METEOR* quantitative metrics. All metric scores are computed between original and reconstructed sentences on the held-out test set to avoid any kind of bias. Table 4 summarizes the overall semantic similarity and lexical overlap results on the held-out test set. The average cosine similarity is high (0.87), and the average GPT-4o (GPT) score is 77.63, indicating very semantically similar sentence reconstruction. As discussed in Section 3.7.3, the results demonstrate strong signs of semantic preservation. Lexical overlap metrics such as ROUGE-1 F1 (0.57) and METEOR (0.49) show partial lexical structure retention, while BLEU (0.15) remains relatively low due to paraphrasing, clause restructuring, reordering of words, and rewording with semantically similar alternative terms.

Table 4. Test-set evaluation statistics across semantic and lexical similarity metrics (n = 274), as reported previously in [2].

Statistic	LLM Score Claude	Human Score	BLEU	ROUGE-1 F1	METEOR	Cosine Similarity
Mean	56.89	4.131	0.150	0.573	0.491	0.872
Standard Deviation	16.87	1.024	0.122	0.114	0.141	0.068

The distributions of these metrics are also shown in Figures 2–4.

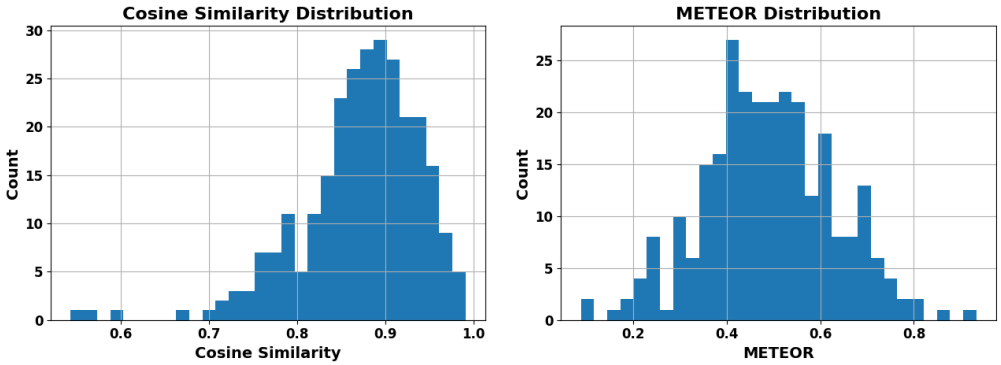


Figure 2. Distribution of Cosine Similarity and METEOR scores across test set reconstructions.

Most *cosine similarity* values are above 0.85, suggesting high semantic retention. *BLEU* scores are relatively low, which can be expected due to lexical variations.

In our analysis, we found that structured representations preserve the meaning of scientific sentences even when lexical overlap is low. The average cosine similarity for the test set (n = 274) is 0.87, showing strong semantic retention. Tables 5–7 show selected reconstructions and their evaluation scores.

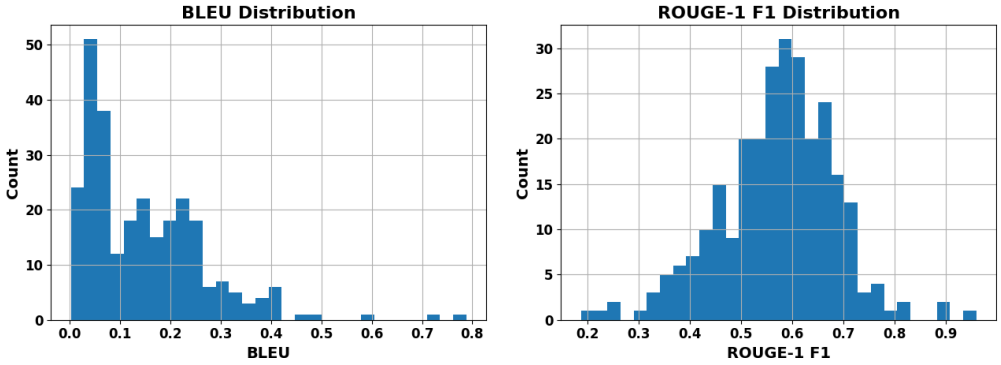


Figure 3. Distribution of BLEU and ROUGE-1 F1 scores across test set reconstructions.

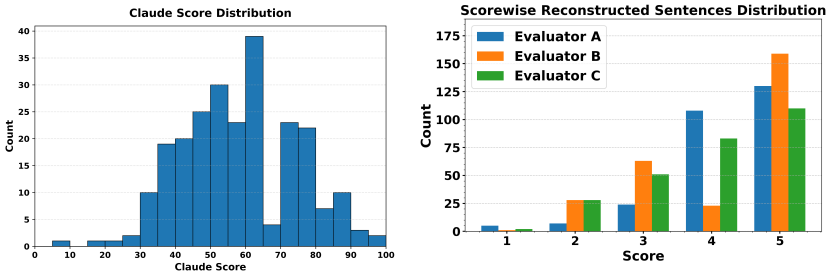


Figure 4. Distribution of LLM-based ratings (**left**) and human evaluator ratings (**right**) across test set reconstructions.

Table 5. Evaluation Scores for Selected Sentences.

Sentence ID	Cosine	BLEU	METEOR	ROUGE	Claude Score	Avg. Human Score
6	0.777	0.051	0.450	0.526	52	3
250	0.824	0.020	0.219	0.311	45	4
415	0.991	0.727	0.871	0.905	82	5
522	0.543	0.002	0.097	0.250	5	1
700	0.975	0.398	0.805	0.776	90	5
873	0.786	0.035	0.3	0.388	25	2.33
1287	0.913	0.015	0.279	0.425	62	4.67

High-scoring examples, such as sentence 415 (*cosine similarity* = 0.99), exhibit near-perfect reconstruction. Similarly, sentence 700 (*cosine similarity* = 0.97) retained full meaning with only minor variation. Failures occur when nested or conditional clauses are not captured correctly. Sentence 6 (*cosine similarity* = 0.77) dropped hedging and attribution, shifting tone. Sentence 250 (*cosine similarity* = 0.82) was compressed, losing context. Sentence 522 (*cosine similarity* = 0.54) was truncated and unusable. Performance is generally consistent in physics, computer science, and medicine. Chemistry and economics show greater variability. This variability may stem from the greater use of domain-specific jargon and complex clause structures in these fields. Sentence 873 (*cosine similarity* = 0.78) demonstrates this with a loss of technical specificity in reconstruction leading to alteration of meaning. We found that BLEU and ROUGE scores remain low across most samples, even when cosine similarity is high. Sentence 1287 shows high cosine (0.913) despite low BLEU (0.015). Sentence 522 shows low scores across all metrics.

Table 6. High Semantic Similarity Examples.

ID	Original Sentence	Reconstructed Sentence
415	We recently reported extremely rapid changes in chromospheric fine structure observed using the IBIS instrument in the red wing of H-alpha.	Extremely rapid changes in chromospheric fine structure
700	The quantum annealer is designed to find the ground state of Ising Hamiltonians with pairwise interactions on a fixed sparse graph called chimera.	The quantum annealer design aims to find the ground state of Ising Hamiltonians, characterized by pairwise interactions, using a fixed sparse graph structure known as chimera.
1287	Recent analyses of aggregated data from Africa have shown that insecticide-treated nets have had the greatest effect on lowering malaria prevalence over the past fifteen years, suggesting that the success of different types of malaria interventions is region specific. Continuing global efforts on malaria elimination should focus on both vector control measures and increased access to artemisinin-combination therapies.	Malaria control strategies, particularly insecticide-treated nets, reduce malaria prevalence more effectively than other interventions, although their success varies by region, suggesting a future focus on vector control and artemisinin therapies for malaria elimination.

Table 7. Medium and Low Semantic Similarity Examples.

ID	Original Sentence	Reconstructed Sentence
6	There is no hypothetical consensus; we believed that it is a multifactorial process that starts with an embryological error from mismigrated cells in addition to mutations of oncogenes and tumor suppressor genes, in which different endocrinological, growth, and vascular factors produce an ideal environment.	Embryological error and cancer development are part of a multifactorial process in cancer, involving combined factors such as mismigrated cells, oncogene mutations, tumor suppressor gene mutations, endocrinological factors, growth factors, and vascular factors.
250	The purpose of this paper was to show that artificial neural networks do have the potential to determine quantitatively the values of the parameters of the underlying theory and the corresponding statistical uncertainties.	Neural networks can estimate parameters by determining them quantitatively, including statistical uncertainties.
522	Therefore, in several cases they perform poorly compared to the fully digital approaches since they can not exploit the inherent spatial multiplexing gain of the channel.	In the comparison of digital vs. (Incomplete construction)
873	One might attempt to rescue such models by adding in small fluctuations away from equilibrium, but that could be done only if the equilibrium could be claimed to be stable, or if it could be claimed to become stable via some sort of coarse-graining.	Small fluctuations may rescue models if small fluctuations are needed and the equilibrium must be stable, allowing the equilibrium to become stable.

Inter-rater agreement in Human Evaluation: In human evalution, agreement was strongest for high-quality reconstructions scored as 4 or 5. Evaluator pairs 1–2, 1–3, and 2–3 jointly assigned scores within this range to 133, 166, and 175 of the 274 samples, respectively. Exact pairwise agreement occurred for 116, 123, and 133 samples. The linear weighted Cohen's kappa values are 0.214, 0.260, and 0.265; Fleiss' kappa is 0.174; and Krippendorff's alpha is 0.313. The moderate variation reflects the inherently subjective nature of natural-language quality evaluation.

4.3. Comparison with the Baseline SRO Representation

In order to assess whether the observed reconstruction performance is specifically attributable to the hierarchical representation, we introduced a compact flat relational baseline for direct comparison. The flat Subject–Relation–Object (SRO) representation was generated from each original scientific sentence using GPT-4o and a dedicated baseline prompt. The prompt required exactly one JSON triple containing only `subject`, `relation`, and `object`. Each field was limited to 30% of the original sentence length to enforce compression comparable to the hierarchical JSON representation and to prevent the baseline from preserving information merely through longer text fields. The generated SRO triples were passed through the same sentence-reconstruction and evaluation pipeline used for the hierarchical JSON representations. In Table 8 we report the mean and standard deviation for BLEU, METEOR, ROUGE-1 F1, cosine similarity, word-edit distance, and Claude judge score for both representation methods. It can be observed in the table the reconstruction from hierarchical representation is better both lexically and semantically. We also reported Paired Wilcoxon signed-rank test results in Table 8. Based on the p-values, the Hierarchical JSON based reconstruction results are significant ($p < 0.05$) on all the metrics.

Table 8. Comparison between baseline SRO triples and hierarchical JSON representations. Values are mean ± standard deviation.

Method	BLEU	METEOR	ROUGE-1 F1	Cosine Similarity	Word-Edit Distance	Claude Judge
Baseline SRO	0.03 ± 0.08	0.26 ± 0.13	0.41 ± 0.14	0.76 ± 0.13	0.74 ± 0.12	34.97 ± 16.05
Hier. JSON	0.13 ± 0.14	0.49 ± 0.14	0.57 ± 0.11	0.87 ± 0.07	0.52 ± 0.09	56.89 ± 16.87
Wilcoxon statistic	32,896	36,827	34,954	33,707	737.5	31,365.5
Wilcoxon	$4.66\times$	$4.91\times$	$6.11\times$	$4.88\times$	$2.30\times$	$1.04\times$
p-value	10^{-27}	10^{-43}	10^{-35}	10^{-30}	10^{-43}	10^{-33}
$p < 0.05$	Yes	Yes	Yes	Yes	Yes	Yes

4.4. t-SNE-Based Visual Analysis in Projected Embedding Space

While the summary metrics provide an overall quantitative assessment, they do not show how individual original–reconstruction pairs are positioned relative to each other. Therefore, we use t-SNE to provide a direct visual inspection of pair-level shifts in the embedding space. We used random initialization, auto learning rate, and a perplexity parameter proportional to the group size (capped at 30 and reduced for small groups to avoid degenerate configurations. For our usual group size of 40, we used 13, and for a lower similarity group size of 34, we used 11 as the perplexity parameter value). To examine the correlation between visualization and semantic quality, we stratify the data by GPT-4o score into two bands: 0–60 ("low similarity") and 61–100 ("high similarity"). Within each band, examples are sorted by their score and assigned a running pair index. For the low similarity group, we get up to 34 sentence pairs, which we marked as Group 1, and for the remaining group, we get up to 240 pairs in total. These 240 pairs are then randomly split into six blocks of 40 pairs to keep each figure comprehensible, creating Groups 2 to 7 in our plots. For every block (along with the low-score group) we construct a t-SNE projection

over all originals and reconstructions in that subset, so that embedding distances within a panel are comparable as well as visually presentable.

The resulting plots use a consistent visual encoding for each pair. Each original sentence is shown as a filled circle, its reconstruction as a cross, and we also plot a faint circular bubble centered at the midpoint between the two points. All three markers associated with a pair share the same color, taken from a high-contrast, perceptually distinct palette generated in the CIELab space (with hues near pure red excluded to reserve red for other uses). The radius of the bubble is kept proportional to the Euclidean distance between the original and reconstructed embeddings in the 2D t-SNE space, so larger bubbles represent a greater semantic displacement. The axes are labeled "Dimension 1" and "Dimension 2", and the aspect ratio is fixed to preserve geometric relationships.

The visualization in Figures 5–11 allows us to directly inspect how the quality of reconstruction relates to geometry in the learned semantic space. Pairs with high-quality reconstruction appear as nearly overlapping circles and crosses with small bubbles, while lower-quality reconstructions appear as more widely separated markers and larger bubbles. We can see from the plots how far individual reconstructions move in embedding space, but also how these movements cluster or disperse across different semantic similarity regions. By comparing the lower- and high-similarity panels, we observed that shifts from the original sentences were more frequent in the low-similarity group. However, it should be noted, sentence content, subject matter, and reconstruction quality may also influenced movement in the t-SNE space.

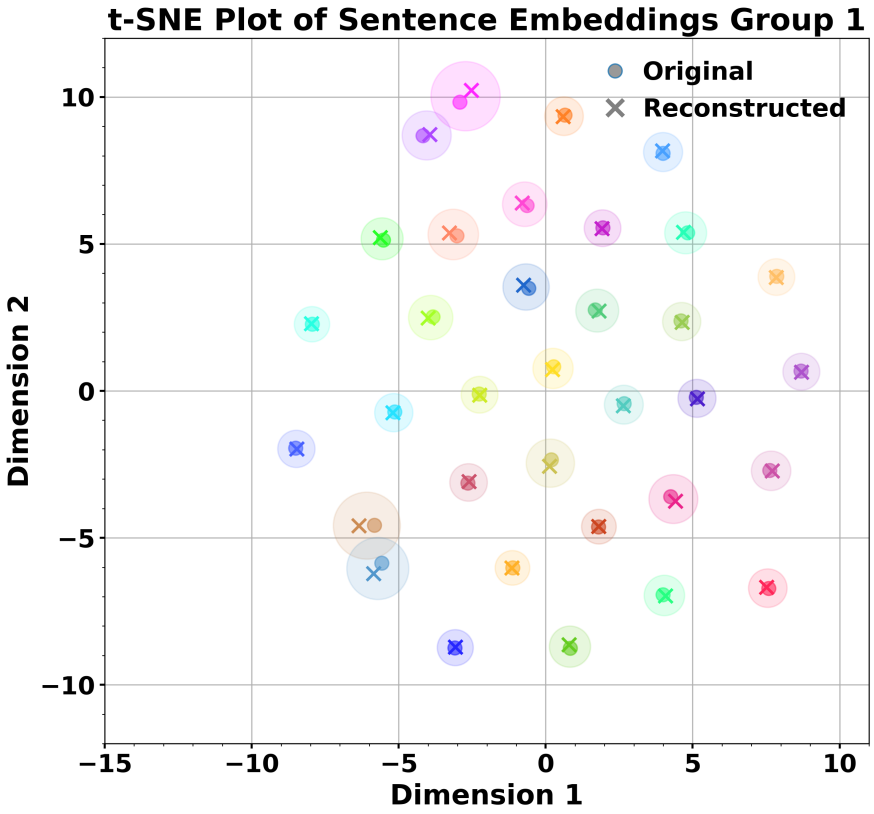


Figure 5. Plot of Group 1 (Low Similarity Group) of original and reconstructed sentence embeddings points pair projected in 2D t-SNE space and their distance represented by proportionate bubble.

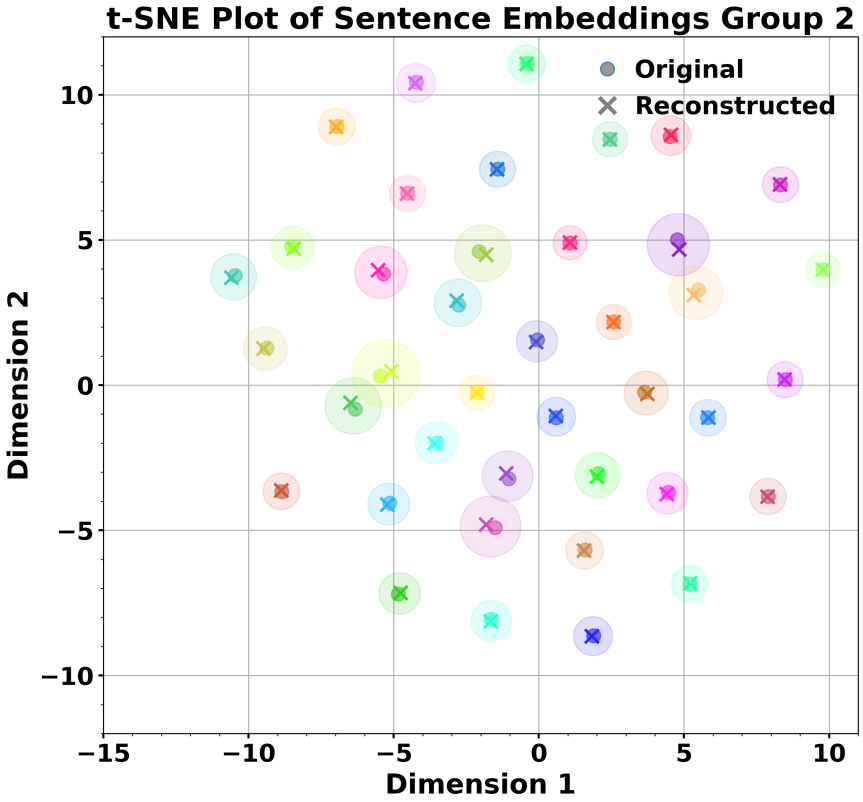


Figure 6. Plot of Group 2 of original and reconstructed sentence embeddings points pair projected in 2D t-SNE space and their distance represented by proportionate bubble.

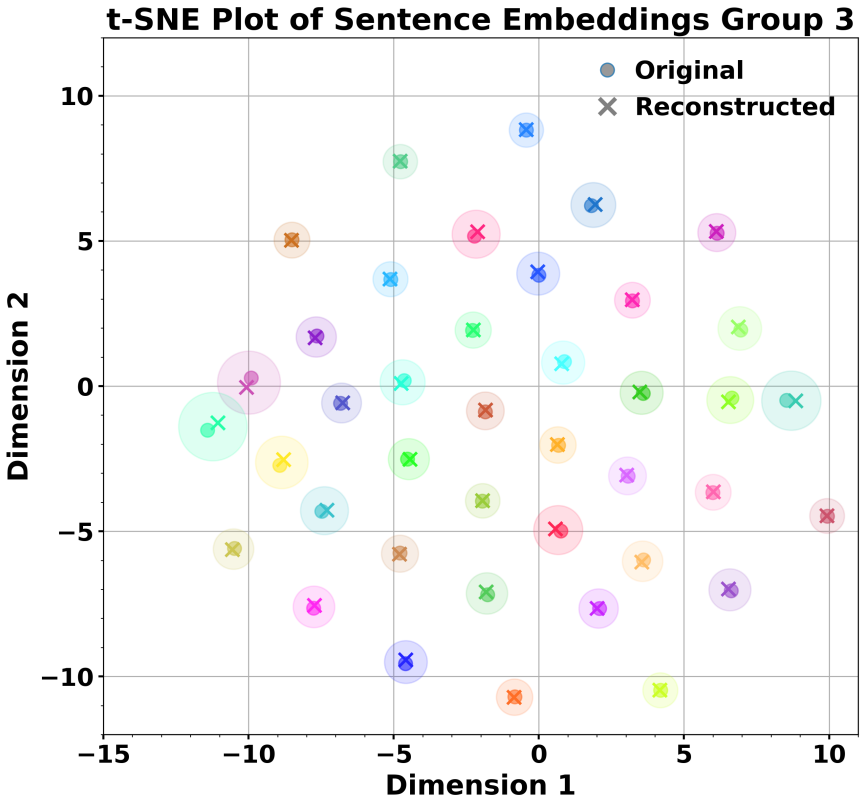


Figure 7. Plot of Group 3 of original and reconstructed sentence embeddings points pair projected in 2D t-SNE space and their distance represented by a proportionate bubble.

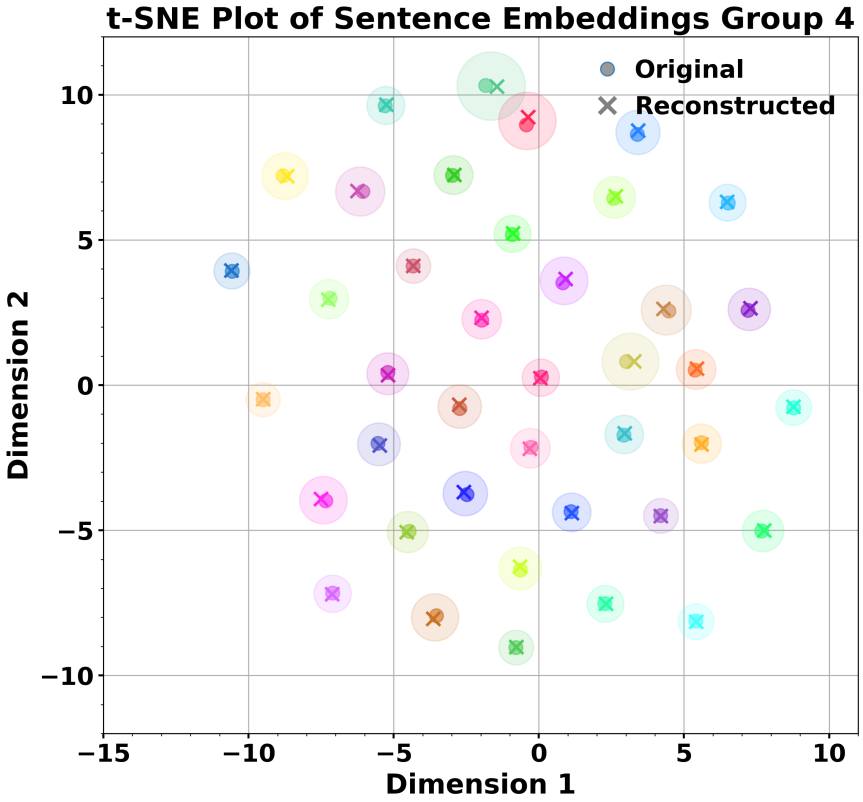


Figure 8. Plot of Group 4 of original and reconstructed sentence embeddings points pair projected in 2D t-SNE space and their distance represented by a proportionate bubble.

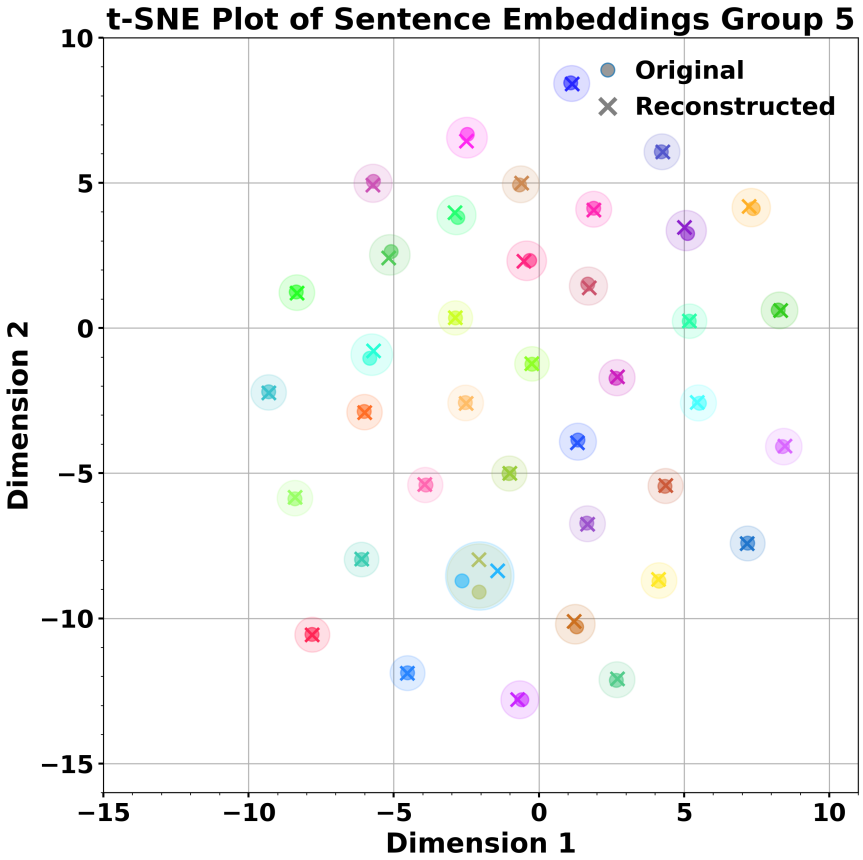


Figure 9. Plot of Group 5 of original and reconstructed sentence embeddings points pair projected in 2D t-SNE space and their distance represented by a proportionate bubble.

https://doi.org/10.3390/bdcc10090288

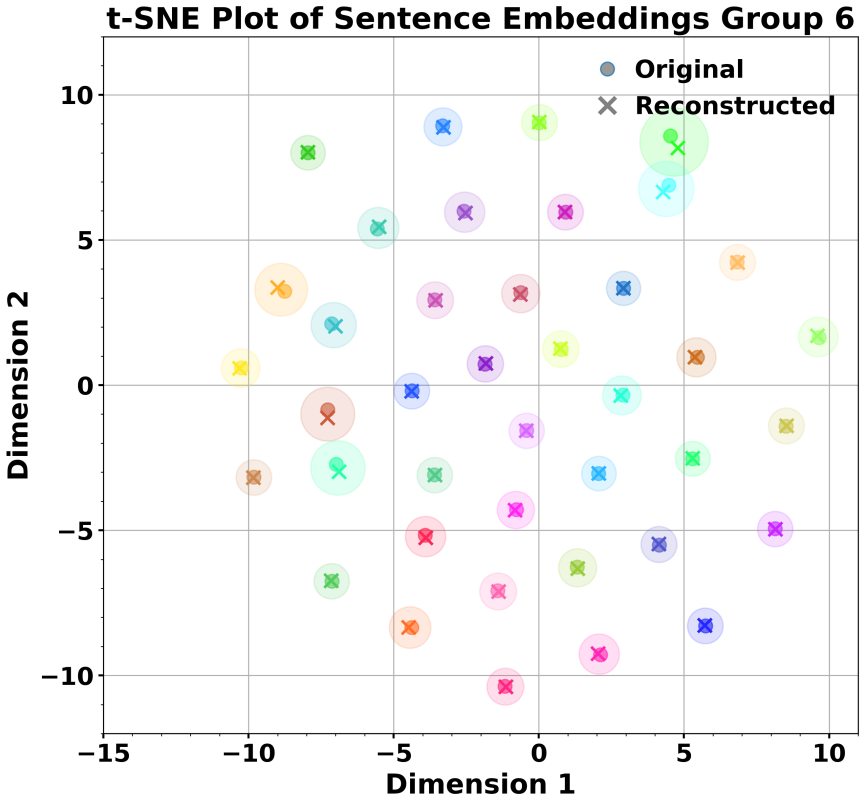


Figure 10. Plot of Group 6 of original and reconstructed sentence embeddings points pair projected in 2D t-SNE space and their distance represented by proportionate bubble.

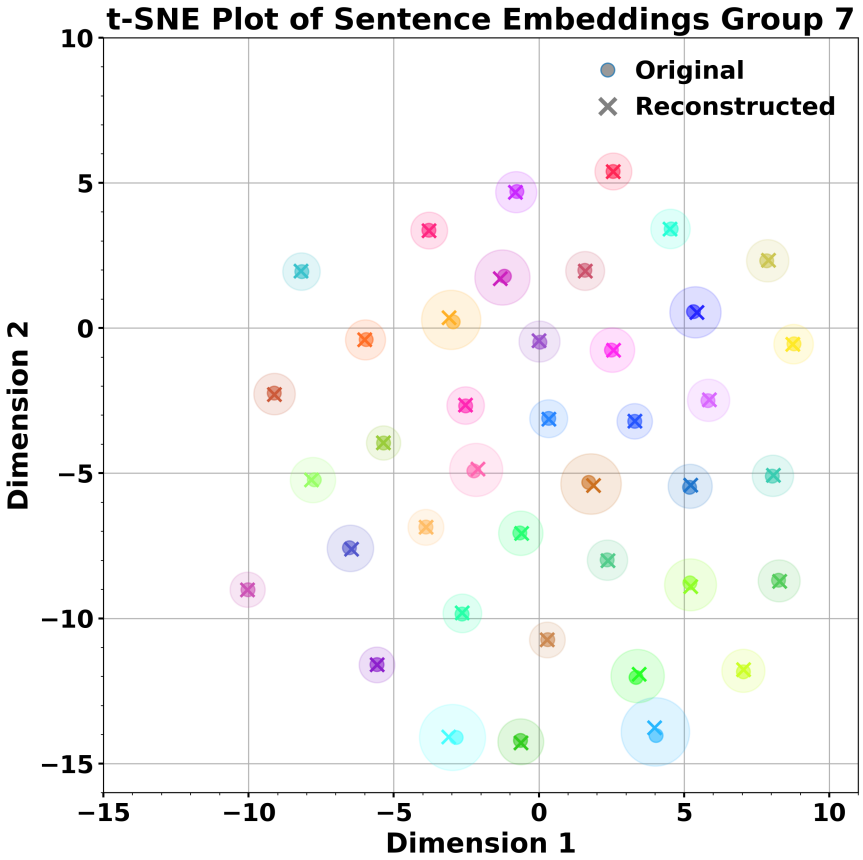


Figure 11. Plot of Group 7 of original and reconstructed sentence embeddings points pair projected in 2D t-SNE space and their distance represented by proportionate bubble.

https://doi.org/10.3390/bdcc10090288

4.5. Final Assessment

Our study finds that the JSON-based structure is quite effective in retaining semantic information. Most reconstructions score above 0.85, confirming general semantic retention. The failures that we observed were due to structural issues or oversimplifications. Our manual inspection also found that even in longer and more complex sentences, the meaning was preserved in the reconstructed sentence.

In Table 9, we show two such interesting examples. The reconstruction of Sentence 16 was precise and well paraphrased, without losing any information. Sentence 183 also did not lose original scientific meaning, though the structure and emphasis were slightly different.

Table 9. Examples of meaning-preserving Sentence Reconstruction.

Sentence ID: 16
Original: Evaluation of antimicrobial activity shows that several compounds exhibit good activity when compared with the reference drug candidates and thus could be promising new lead molecules.
Reconstructed: The evaluation of antimicrobial activity reveals that the compounds show promising antimicrobial activity compared to reference drugs, suggesting their potential as new lead molecules.
Cosine: 0.964 *BLEU:* 0.149 *METEOR:* 0.520
Sentence ID: 183
Original: Researchers proposed a method for solid mercury where the mean field part is calculated for the infinite solid and the correlation part is obtained through many body expansion.
Reconstructed: The method proposed for solid mercury, composed of a mean field part and a correlation part, involves calculating the mean field part and obtaining the correlation part, which is applied to an infinite solid and expanded through a many-body expansion.
Cosine: 0.944 *BLEU:* 0.132 *METEOR:* 0.649

5. Conclusions

This study investigated whether hierarchical structured JSON representations can preserve the meaning of scientific sentences. The results suggest that minimal, well-formed structure can serve as an effective intermediate format (akin to stored vector embeddings of sentences) for capturing scientific ideas. The structured outputs indicate that structured representation may be beneficial for interpretability and semantic comparison. These kinds of structural outputs can also be seamlessly integrated into the multi-agentic setups where agents expect a structured format.

During fine-tuning, we currently use the standard cross-entropy loss. In addition, we apply a penalty when the model output is not valid JSON. During training, each output is decoded and checked. If it cannot be parsed as JSON, the loss increases. This strategy encourages correct syntax but does not account for structure or semantic accuracy. In future work, we plan to use a tree or graph-based comparison for this task. Loss derived from such comparisons can match predicted and true outputs by structure and keys. A tree or graph-based comparison strategy can help evaluate both syntactic form and semantic content more effectively.

Future work may include validation using larger and more diverse datasets to further assess generalizability. Reconstruction failures may also be systematically categorized, including missing conditions, loss of causal relationships, truncation, and incorrect entity substitution. The effects of incomplete or corrupted structures, missing fields, incorrect labels, and deeper or more abstract hierarchies may likewise be examined in future. The use of a single representative 7B-parameter model is also a limitation of this study.

Although similar-sized models may provide comparable structured-generation capabilities, this should be verified through future cross-model comparisons.

Author Contributions: Conceptualization, A.M.; Methodology, A.M., N.S., S.S.R.N. and A.J.; Software, A.J.; Validation, S.S.R.N. and E.Y.; Formal analysis, S.S.R.N. and E.Y.; Investigation, S.S.R.N., E.Y. and N.S.; Resources, A.J. and A.M.; Data curation, S.S.R.N. and E.Y.; Writing—original draft, A.M., N.S. and A.J.; Writing—review & editing, A.M. and A.J.; Visualization, N.S.; Supervision, A.M. and A.J.; Funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the NASA Established Program to Stimulate Competitive Research (EPSCoR), Grant No. 80NSSC22M0027.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The input sentences used in this study are publicly available at https://github.com/ani-maiti/information_retentation_json (accessed on 17 July 2026).

Acknowledgments: This research was partially supported by the NASA Established Program to Stimulate Competitive Research (EPSCoR), **Grant #80NSSC22M0027**, through the Seed Grant awarded to Aniruddha Maiti by the NASA West Virginia EPSCoR Committee for the period 1 June 2025–31 May 2026. Part of the computational work for this research was supported by resources at the *Pittsburgh Supercomputing Center (PSC)* and *Anvil at Purdue University* through ACCESS allocations **CIS250884 (Campus Champion)** and **CIS251319 (Campus Champion)**. ACCESS is supported by the U.S. National Science Foundation under grants #2138259, #2138286, #2138307, #2137603, and #2138296. Local computing resources required for this research were provided through a workstation funded by the WV Innovation Grant ING25-005 awarded to Aniruddha Maiti. **AI tool use:** The authors acknowledge that AI-based tools were used solely for grammar correction and to improve the clarity and readability of the manuscript, with all such revisions thoroughly verified by the authors. All scientific content, methodology, analyses, results, interpretations, and conclusions are the authors' original work.

Conflicts of Interest: Author Niladri Sengupta was employed by the Fractal Analytics Inc.. The remaining authors declare no conflict of interest.

References

1. Maiti, A.; Adewumi, S.; Tikure, T.A.; Wang, Z.; Sengupta, N.; Sukhanova, A.; Jana, A. Comparative Analysis of OpenAI GPT-4o and DeepSeek R1 for Scientific Text Categorization Using Prompt Engineering. *arXiv* **2025**, arXiv:2503.02032.
2. Nimmagadda, S.S.R.; Young, E.; Sengupta, N.; Jana, A.; Maiti, A. Generating Hierarchical JSON Representations of Scientific Sentences Using LLMs. In *Proceedings of the 2026 International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 2–4 February 2026*; IEEE: Piscataway, NJ, USA, 2026; pp. 218–220.
3. Hsu, T.Y.; Giles, C.L.; Huang, T.H. SciCap: Generating captions for scientific figures. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7–10 November 2021; pp. 3258–3264.
4. Yang, Z.; Dabre, R.; Tanaka, H.; Okazaki, N. Scicap+: A knowledge augmented dataset to study the challenges of scientific figure captioning. *J. Nat. Lang. Process.* **2024**, *31*, 1140–1165. [CrossRef]
5. Yue, T.; Shi, X.; Mao, R.; Song, Z.; Hu, Z.; Zhao, D. AnaFig: A Human-Aligned Dataset for Scientific Figure Analysis. In Proceedings of the 33rd ACM International Conference on Multimedia, Dublin, Ireland, 27–31 October 2025; pp. 12837–12843.
6. Shukla, S.; Sharma, N.; Gupta, M.; Mishra, A. Patentlmm: Large multimodal model for generating descriptions for patent figures. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PN, USA, 25 February–4 March 2025; Volume 39, pp. 20488–20496.
7. Zhang, L.; Eger, S.; Cheng, Y.; Zhai, W.; Belouadi, J.; Leiter, C.; Ponzetto, S.P.; Moafian, F.; Zhao, Z. ScImage: How good are multimodal large language models at scientific text-to-image generation? *arXiv* **2024**, arXiv:2412.02368.
8. Belouadi, J.; Ilg, E.; Keuper, M.; Tanaka, H.; Utiyama, M.; Dabre, R.; Eger, S.; Ponzetto, S. Tikzero: Zero-shot text-guided graphics program synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Honolulu, HI, USA, 19–25 October 2025; pp. 17793–17806.

9. Liang, C.; You, J. DiagramEval: Evaluating LLM-Generated Diagrams via Graphs. *arXiv* **2025**, arXiv:2510.25761.
10. Song, C.; Raghunathan, A. Information leakage in embedding models. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 9–13 November 2020; pp. 377–390.
11. Morris, J.X.; Kuleshov, V.; Shmatikov, V.; Rush, A.M. Text embeddings reveal (almost) as much as text. *arXiv* **2023**, arXiv:2310.06816.
12. Seputis, D.; Li, Y.; Langerak, K.; Mihailov, S. Rethinking the Privacy of Text Embeddings: A Reproducibility Study of "Text Embeddings Reveal (Almost) As Much As Text". In Proceedings of the Nineteenth ACM Conference on Recommender Systems, Prague, Czech Republic, 22–26 September 2025; pp. 822–831.
13. Cheng, J.; Reddy, S.; Saraswat, V.; Lapata, M. Learning structured natural language representations for semantic parsing. *arXiv* **2017**, arXiv:1704.08387.
14. Wang, X.; Bai, X. Entity-Aware Self-Attention and Contextualized GCN for Enhanced Relation Extraction in Long Sentences. *arXiv* **2024**, arXiv:2409.13755.
15. Liu, R.; Yin, R.; Bo, X.; Hao, X.; Liu, Y.; Zhong, J.; Ma, C.; Wang, W. SSTAG: Structure-Aware Self-Supervised Learning Method for Text-Attributed Graphs. *arXiv* **2025**, arXiv:2510.01248.
16. Zheng, C.; Chen, X.; Xu, R.; Chang, B. A double-graph based framework for frame semantic parsing. *arXiv* **2022**, arXiv:2206.09158.
17. Sun, Y.; Huang, Q.; Tung, A.K.; Yu, J. PRISM: A Framework for Producing Interpretable Political Bias Embeddings with Political-Aware Cross-Encoder. *arXiv* **2025**, arXiv:2505.24646.
18. Das, S. HyperSumm-RL: A Dialogue Summarization Framework for Modeling Leadership Perception in Social Robots. In Proceedings of the 36th ACM Conference on Hypertext and Social Media, Chicago, IL, USA, 15–18 September 2025; pp. 171–176.
19. Thilagam, P.S.; Uma, P.D. Leveraging Structural and Semantic Measures for JSON Document Clustering. *J. Univers. Comput. Sci. (JUCS)* **2023**, *29*, 222. [CrossRef]
20. Yang, J.; Jiang, D.; He, L.; Siu, S.; Zhang, Y.; Liao, D.; Li, Z.; Zeng, H.; Jia, Y.; Wang, H.; et al. StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs. *arXiv* **2025**, arXiv:2505.20139.
21. Woof, W.; Chen, K. A framework for end-to-end learning on semantic tree-structured data. *arXiv* **2020**, arXiv:2002.05707.
22. Shorten, C.; Pierse, C.; Smith, T.B.; Cardenas, E.; Sharma, A.; Trengrove, J.; van Luijt, B. Structuredrag: JSON response formatting with large language models. *arXiv* **2024**, arXiv:2408.11061.
23. Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; Wei, F. Improving text embeddings with large language models. *arXiv* **2023**, arXiv:2401.00368.
24. Dunn, A.; Dagdelen, J.; Walker, N.; Lee, S.; Rosen, A.S.; Ceder, G.; Persson, K.; Jain, A. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv* **2022**, arXiv:2212.05238.
25. Dagdelen, J.; Dunn, A.; Lee, S.; Walker, N.; Rosen, A.S.; Ceder, G.; Persson, K.A.; Jain, A. Structured information extraction from scientific text with large language models. *Nat. Commun.* **2024**, *15*, 1418. [CrossRef] [PubMed]
26. Shahriar, S.; Lund, B.D.; Mannuru, N.R.; Arshad, M.A.; Hayawi, K.; Bevara, R.V.K.; Mannuru, A.; Batool, L. Putting gpt-4o to the sword: A comprehensive evaluation of language, vision, speech, and multimodal proficiency. *Appl. Sci.* **2024**, *14*, 7782. [CrossRef]
27. Ahmad, A.; Kowol, P.; Hillmann, S.; Möller, S. Multi-intent recognition in dialogue understanding: A comparison between smaller open-source LLMs. *arXiv* **2025**, arXiv:2509.10010.
28. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PN, USA, 6–12 July 2002; pp. 311–318.
29. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
30. Banerjee, S.; Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72.
31. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *ICLR* **2022**, *1*, 3.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.