Keyword-guided structured abstract generation for deep learning papers using ChatGPT-4o

CELIA MEDINA GIMENEZ, University of Twente, The Netherlands

The exponential growth of deep learning research has made it difficult for publishers and readers to write and evaluate scientific papers in an efficient way. While automatic summarisation tools exist, they normally generate brief, unstructured outputs and lack transparency or content validation mechanisms. This project introduces a framework that generates IMRAD-structured abstracts using OpenAI's GPT-4o, guided by extracted keywords and evaluated through both automatic and Large Language Model (LLM) based methods. The system incorporates a keyword validation loop that enforces the inclusion of the most important concepts and iteratively improves abstract quality. Evaluation is performed using semantic similarity metrics, natural language inference (NLI), and judgment from two independent LLMs (Gemini and Claude), each rating factual accuracy, clarity, completeness, and keyword relevance. Results show that the proposed framework improves factual consistency, coverage, and semantic alignment over a simple prompt baseline, though may introduce trade-offs in clarity. These findings demonstrate the value of structured prompting, and keyword feedback in scientific summarisation.

Additional Key Words and Phrases: structured abstract generation, keyword validation loop, iterative prompting, large language models, scientific summarisation, factual alignment, IMRAD structure, deep learning.

1 INTRODUCTION

In the last years, there has been an exponential growth in scientific publications in the field of computer science, particularly deep learning. In publishing platforms like arXiv, AI-related categories, including cs.LG and stat.ML, have experienced a sudden number increase doubling approximately every 23 months due to the continuous advancements in the field [14]. Publishers then face the challenge to create large volumes of scientific content. Consequently, the need for automatic tools to guide and support publishers arises.

It is important to note that the AI generation of most of the paper's written content is generally discouraged because of concerns about originality and academic integrity [26]. However, the nature of the abstract and keywords section differs, as they aim to summarise and highlight the work's main points, which are based on the rest of the already completed work. Therefore, if the correct framework is applied, the generated output can be reliable and seamlessly integrated with the rest of the paper.

Currently, tools like SciSummary provide quick summaries in a unstructured, single paragraph format [1]. These summaries have shown inconsistent coverage of key points and often produce outputs that are too generic or too focused on one section of the paper. This limitation makes it difficult to extract key findings or methods quickly, especially in complex papers. In technical fields, such as deep learning, sections like methodology are essential and need to be correctly represented and explained, which makes this kind of summarisation insufficient.

A scientific abstract should give a clear, high-level understanding of a paper's main results and contributions. It has been shown that structured abstracts are easier to read and navigate and improve information retrieval, reader comprehension, and scientific communication [10]. More specifically, the IMRAD format is commonly used in scientific writing and follows: Introduction, Methods, Results and Discussion. It provides a logical, familiar, and standard way of presenting scientific research [28]. Despite these benefits, it is rarely automated in these types of tools.

Similarly, the role of keywords have been overlooked. Keywords help with indexing and searchability in digital libraries, and informs readers about the content focus [27]. However, current summarisation tools ignore them completely or fail to ensure their inclusion in the generated summaries, leaving them unrepresented. Moreover, their potential role in abstract generation remains to be unexplored. Since keywords summarise the core concepts of a paper, they could be introduced in the abstract generation and guide Large Language Models (LLMs) in the process.

At present, no available tool focuses specifically on deep learning papers, generates structured abstracts and integrates keywords as part of both the input and the quality control process. This project introduces an iterative keyword-based framework that aims to guide and validate abstract generation, where extracted keywords are used both to prompt the LLM and to assess whether the generated abstract covers core concepts, making it revise if key terms are missing and therefore addressing the limitations in the existing solutions.

Ensuring the factual accuracy of the generated content is one of the main challenges in the study. As LLMs may omit or misrepresent [2], this research emphasises the importance of an evaluation mechanism to detect incomplete or unrelated summaries. The keyword validation step serves as a core method to reduce hallucinations and improve the alignment with the source paper.

The objective of this research is to build an abstract generation framework for scientific publishers, tailored to deep learning papers. The central hypothesis is that structured prompting combined with keyword guidance can improve the quality of LLM-generated abstracts in factual accuracy, completeness, and relevance. To address this hypothesis, the following research and sub-research questions will be answered.

RQ: To what extent can structured keyword-guided prompts improve the factual accuracy, coverage, and relevance of LLM-generated abstracts for deep learning papers?

This can be answered through the following sub-research questions (SRQ's):

Author's address: Celia Medina Gimenez, c.medinagimenez@student.utwente.nl, University of Twente, P.O. Box 217, Enschede, The Netherlands, 7500AE.

TScIT 43, July 8, 2022, Enschede, The Netherlands

SRQ1: How does the quality of abstracts generated with the proposed structured keyword-guided framework compare to those generated with a flat, single-prompt baseline?

SRQ2: To what extent does the iterative keyword validation loop improve abstract quality across successive generation attempts within the framework?

2 RELATED WORK

Scientific article summarization is gaining attention as researchers seek to represent complex studies into clear and accurate abstracts. Among scientific writing standards, the IMRAD structure (Introduction, Methods, Results, Discussion) has become a widely used and well-supported format for both authors and readers [10, 28]. Many summarization tools use this structure because it improves readability and makes use of academic writing structure[23]. However, IMRAD is a structure, it is not in itself a novel mechanism for content generation or validation.

Recent frameworks have applied deep learning models and large language models (LLMs) to produce scientific summaries, but most rely on a single prompt generation and offer little control over content selection or factual alignment. For example, Oh et al. [23] proposed a method for generating structured abstracts from full-text papers using section-wise summarisation. While this improves distribution across abstract sections, the process still does not have any mechanism for verifying whether key domain concepts are included.

More recent LLM-based systems like SummIt [30] and Self-Refine [21] introduce iterative generation strategies, prompting the model to critique and revise its own outputs. These approaches have shown improvements in fluency and factual accuracy by treating summarisation as a multi-step reasoning process. However, they do not incorporate any external anchors, such as specific keywords, and do not enforce alignment between the abstract and the core scientific content. Other systems like ISQA [18] use fact-checking through LLMs during summarisation, but again without structured guidance or semantic constraints.

In addition, controllable summarisation methods aim to influence what is included in the summary, often through keywords. For example, Li et al. [16, 17] demonstrated that summaries guided by extracted or user-specified key phrases can improve relevance. CTRLsum [3] extended this idea to include a customisable focus to the summaries using keywords as a control method. These works claim that keywords can guide generation effectively, but they treat keyword use only as an input, not a feedback mechanism. If keywords are missing from the generated summary, the model is not prompted to revise its output. Moreover, keyword conditioning is rarely applied in technical or scientific domains, and even less in structured outputs like abstracts.

In summary, existing work on scientific summarisation tends to either follow the IMRAD structure without mechanisms for verifying content coverage, or use keywords in a simple, single-shot way that lacks validation. It seems no existing framework combines structured abstract generation with keyword-based guidance and quality control.

This thesis addresses that gap by introducing a novel framework that generates structured abstracts using an LLM with section-specific prompts, conditions generation on author-defined keywords as semantic anchors, and implements a keyword validation loop that detects missing terms and re-prompts the model to revise the abstract accordingly.

3 RESEARCH METHODOLOGY

This study presents a framework that generates structured abstracts from scientific papers using large language models (LLMs), guided by keywords, and evaluates it through various methods. The approach has five main phases: data extraction and preprocessing, IMRAD and keyword extraction, structured abstract generation, keyword-based validation, and evaluation. This section explains the motivation and technical implementation of each stage.

3.1 Overview of the Pipeline

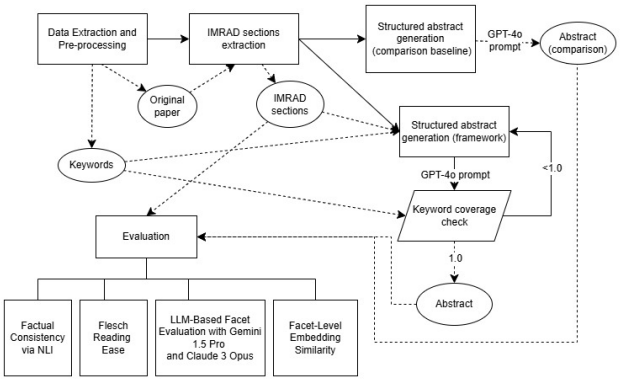

Fig. 1. Pipeline

The pipeline consists of the following phases:

(1) **Data Extraction and Pre-processing**: Cleans raw papers by removing irrelevant sections, giving a plain text input.
(2) **IMRAD and Keyword extraction**: Uses regular expressions to extract the Introduction, Methods, Results, and Discussion (IMRAD) sections from cleaned papers, along with their keywords.
(3) **Structured Abstract Generation**: Prompts an LLM (ChatGPT-4o) to generate abstracts in IMRAD format, conditioned on both the content of the paper and the keywords.
(4) **Keyword-Based Validation Loop**: Automatically checks if all keywords are included and prompts the model again if any are missing.
(5) **Evaluation**: Assess abstract quality using different metrics.

3.2 Pre-processing techniques

Unlike open repositories such as arXiv, which offer easier extraction pipelines due to the availability of LaTeX source files, this project uses scientific papers from the SpringerLink digital library. Springer publications were selected because they consistently include clearly

labelled keyword sections, a critical feature for this framework's generation and evaluation process. In contrast, arXiv papers often lack keywords, making them less suitable for this study. This preprocessing step produced a clean, structured version of each paper.

3.3 IMRAD and Keyword Extraction

To guide the abstract generation, each cleaned scientific paper is divided into its different sections according to the IMRAD structure: Introduction, Methods, Results, and Discussion. This division is done using regular expression (regex) patterns designed to match a range of common section headings found in academic writing. The regex takes into account most common variants and formatting inconsistencies ("Materials and Methods", "Conclusion(s)", or numbered headers like "3. Methods") to identify and extract each section's content.

This structured segmentation is motivated by findings that structured abstracts improve readability, completeness, and information retrieval performance, particularly in biomedical and scientific domains [6].

In parallel, keywords are extracted directly from the body of each paper. Instead of relying on LLMs or statistical models, this approach uses regex patterns to obtain author-defined keyword blocks.

The resulting IMRAD sections and keyword list are used as inputs for the next phase of the pipeline.

3.4 Structured Abstract Generation

This step generates a structured abstract following the IMRAD format with OpenAI's `gpt-4o`. Each abstract is created with the information of the previously extracted IMRAD sections and a keyword list.

The prompt defines the output format to have four labelled sections: Introduction, Methods, Results, and Discussion. Each section is generated based on the corresponding source section from the full paper, ensuring that the model focuses on the correct and relevant information. This approach differs from traditional prompt designs, which provide the entire paper to the model as a single flat input.

In addition, the prompt includes semantic anchors: keywords defined by the paper's author. These keywords are mentioned to be in specific sections (domain terms in the Introduction, technical descriptors in the Methods, and broader implications in the Discussion). This design makes it more likely that the generated abstract keeps key concepts and terminology from the original work. This prompt focused on format and keywords aligns with works showing that format conditioning, when combined with content anchors, improves the informativeness and structure of LLM outputs [9, 32]

The generation step is iterative; the system checks whether the abstract covers all extracted keywords. If not, the model is re-prompted using a modified version of the original instruction, explicitly requesting the inclusion of the missing terms. This validation loop continues for a fixed number of attempts (5) or until all the keywords are included semantically.

This methodology ensures that the abstract is organised and based on both the content and vocabulary of the source paper. The output is a structured abstract in paragraph form, divided by section labels, which is then passed to the evaluation phases.

3.5 Evaluation

The quality of the generated abstracts is evaluated using a combination of automatic metrics and semantic analysis. This approach is designed to measure surface-level similarity and also deeper factual alignment between the generated output and the source material.

Facet-Level Embedding Similarity. Each IMRAD section of the generated abstract is compared to its corresponding source section using Sentence-BERT (SBERT) embeddings. This comparison follows approaches used in scientific summarisation research [8, 20], and shows which parts of the structure are more or less represented.

Factual Consistency via NLI. Following recent best practices in evaluating factual grounding [11, 12], a natural language inference (NLI) model is applied. This step categorises statements as entailed, contradictory, or neutral, highlighting factual differences that may not have been captured by embedding similarity alone.

Flesch Reading Ease. The Flesch Reading Ease Score is used to measure the readability of generated abstracts. This metric captures how easy a text is to understand, based on the length and complexity of its sentences [7]. Higher scores indicate simpler, more readable text, and lower scores suggest denser, more technical language. Since abstracts are often the first text read in papers, they must be understandable and easy to read. This is important because low readability in abstracts negatively affects accessibility and reproducibility [25]. Thus, the Flesch score provides an additional metric to evaluate if the generated outputs are usable and communicative.

LLM-Based Facet Evaluation via Gemini and Claude. Google's Gemini 1.5 Pro model and Claude 3 are used to evaluate each IMRAD section based on four criteria: factual accuracy, clarity, structural completeness, and keyword relevance.
Gemini has been shown to perform competitively across different evaluation scenarios, with similar results as human judgments on content quality, factuality, and completeness [4, 22, 31].

Claude also shows to be a strong candidate for independent evaluation, especially in cases where neutrality is important, correlating strongly with human judgments [31], making it suitable for evaluating OpenAI-generated abstracts.

Importantly, because Gemini and Claude are independent from the generation model (GPT-4o), their use reduces the risk of model bias and makes the assessment more impartial [24]. Moreover, using both together is beneficial; recent studies recommend using multiple diverse models in an LLM jury to improve evaluation fairness and reduce individual model bias through triangulation [5].

The same structured prompt was used for both Gemini and Claude models for consistency across evaluations. The prompt explicitly defines the four evaluation criteria, and requests a 1–5 integer score for each, along with a short justification. Additionally, the prompt clearly defines the exact meaning of each of the four criteria. The complete prompt is included in Appendix D.

Prompt Design and Best Practices. The prompt design reflects several best practices from recent literature. Although chain-of-thought

(CoT) prompting has been shown to improve reasoning in LLMs, this approach was not implemented, as its benefits are evident in reasoning that requires multiple steps rather than practical judgment tasks. Instead, justifications from the LLMs were implemented, which have been shown to have similarly strong results for evaluation tasks without increasing verbosity unnecessarily [13].

Furthermore, research indicates that specific, detailed rubrics greatly improve LLM scoring consistency. For this reason, the prompt clearly defines what each evaluation dimension means and uses a 5-point integer scale. Using small, explained integer scales avoids common issues with 0–1 floating-point ratings, where models tend to rate toward midpoint values or produce noisy outputs. Studies also show that overly large scales introduce randomness and numeric bias [15].

Section-Wise Evaluation Strategy. Another key design decision was to do different evaluations per section . Each abstract was divided into IMRAD sections (Introduction, Methods, Results, Discussion), and the LLM was asked to assess one section at a time using the corresponding section from the original paper. This approach was motivated by findings that LLMs struggle to maintain focus on long inputs due to the "lost in the middle" effect [19]. By making the input smaller, the load on the model is reduced, and the outputs seem to be more reliable. Fine-grained evaluations of smaller text are more similar to human evaluation behaviour, where reviewers typically look into scientific papers section by section.

In fact, recent tools such as FineSurE take this approach further by evaluating each sentence or unit individually, showing improved correlation with human judgments and the ability to detect subtle inconsistencies [29]. Consequently, the chosen evaluation allows for a precise identification of strengths and weaknesses.

The practical implementation of each of these components is detailed in the following section, Experimental Setup.

4 EXPERIMENTAL SETUP

This part outlines how the proposed methodology was implemented, including data selection, model configuration, prompt design, iteration limits, and evaluation methods.

4.1 Data Collection and Preprocessing

A dataset of 20 deep learning papers is collected from the SpringerLink digital library. The collected papers are downloaded in PDF format and converted to plain text using the PyMuPDF library.

The preprocessing process includes:

- Abstract removal using regular expressions and positional heuristics
- Filtering figure/table captions by detecting numbered references (e.g., "Figure 3", "Table 1")
- Excluding equations, removing lines with a significant amount of numbers and specific formatting
- Truncating documents at headings such as "References", "Bibliography", "Appendix", etc.

4.2 IMRAD and Keyword Extraction

This phase follows the steps described in the methodology section, where each cleaned paper was divided into the IMRAD structure using regular expressions. The same approach was used to extract author-defined keywords.

The full list of regex patterns used for section and keyword identification is included in Appendix A.

4.3 Structured Abstract Generation

To generate structured abstracts, a custom Python pipeline is implemented using OpenAI's `gpt-4o` model via the official API. Each abstract follows the IMRAD structure and is generated section by section using only the corresponding segment of the full paper as input.

The generation prompt for each paper is constructed dynamically. After loading the paper's IMRAD sections from `.json` files, the script combines each section's content with specific instructions and a list of keywords extracted from the original paper.

The LLM was called with the following configuration:

- **Model**: GPT-4o (OpenAI)
- **Temperature**: 0.1
- **Max generation tokens**: 1000
- **System role instruction**: "You are a highly factual scientific abstract generator."

A key feature is the keyword coverage validation loop. The abstract is compared after generating it against the expected keyword list. If any terms are missing, the model is re-prompted using an updated instruction that explicitly lists the missing keywords. This mechanism runs for up to five attempts per paper, terminating early if full keyword coverage is achieved. All iterations are saved independently for analysis.

Post-generation, the section headers are normalised which ensures consistent formatting (converting "Materials and Methods" to "Methods"). The complete structured prompt used for abstract generation is provided in the Appendix B for reference.

Baseline Comparison Generation

To evaluate the benefit of the proposed structured framework, an additional abstract was generated for each paper using a simpler baseline method. Here, the fully cleaned paper text is provided as a flat prompt to the same LLM, without any IMRAD segmentation or structured prompt guidelines. The complete prompt is provided in the Appendix C for reference.

Both the baseline and the framework-generated abstracts were stored in parallel directories for comparison and evaluation.

4.4 Evaluation

All generated abstracts, including iterative outputs (1–5) and baseline comparisons, are evaluated using the following methods:

Semantic Similarity by Section. : This evaluation loads each paper's IMRAD sections previously extracted and compares them to the corresponding generated abstract sections using SBERT embeddings from the `all-mpnet-base-v2` model. Cosine similarity scores are saved per section to allow section analysis.

NLI-Based Factual Consistency. : Using the `ynie/bart-large-snli_-mnli_fever_anli_R1_R2_R3-nli` model, each sentence from a generated abstract is checked against the original section. Inferences

are classified as `entailment`, `neutral`, or `contradiction`. Then, both label counts and confidence-weighted summary scores are calculated.

Flesch Reading Ease. : Readability is measured using the Flesch Reading Ease Score, computed with the `textstat` library. This score shows how easy the abstract is to read, with higher values indicating simpler language. Scores between 30 and 50 typically correspond to college-level texts [7].

LLM-Based Evaluation via Gemini and Claude. : Both evaluators are accessed via their respective APIs using Python. For each paper, both the generated abstract (either from the structured framework or the baseline) and the corresponding IMRAD-parsed section from the original paper are used as inputs.

The evaluation is done using a rubric-based prompt (see Appendix D) that asks the model to rate each section on factual accuracy, clarity, structural completeness, and keyword relevance. Output is returned in strict JSON format to allow automatic parsing.

The evaluation loop is implemented in Python, using the `google.generativeai` SDK for Gemini and the Anthropic API for Claude. For each model and each abstract, four evaluation records were collected (one per IMRAD section).

The LLMs were called with the following configuration:

- **Claude 3 Opus (Anthropic)**
  - **Model**: Claude 3 Opus (20240229)
  - **Temperature**: 0
  - **Max generation tokens**: 1000
  - **System role instruction**: *"You are a fair, accurate scientific writing evaluator."*
- **Gemini 1.5 Pro (Google)**
  - **Model**: Gemini 1.5 Pro
  - **Temperature**: 0
  - **Max generation tokens**: Not explicitly set (default)
  - **System role instruction**: None (Gemini uses user instructions only)

Output Structure. All evaluation outputs are stored in per-paper CSV logs, with columns for paper ID, section, metric type, score, and iteration.

5 RESULTS

In this section, the performance of the keyword-guided structured abstract generation framework is compared to a simple prompt approach. The results are measured using multiple evaluation metrics on accuracy, relevance, completeness, and readability.

The analysis is divided into two parts. First, to compare the overall abstract quality between the framework and the baseline, only the best-performing abstract from the framework's iterative generation attempts (the version with the highest evaluation scores) is selected for each paper. This ensures that the comparison focuses on the framework's full potential.

Second, the keyword-based validation loop is examined to determine if abstract quality raises with each generation. For this, how the evaluation scores evolve across the framework's iterative attempts was investigated: whether they improve, degrade, or remain stable. This stage aims to evaluate the practical value of keyword coverage through repeated generation.

Table 1. Summary of the Metric comparison between framework and comparison abstracts

	Facet Score	Claude	Gemini	Readability	NLI Entailment
Framework	0.7757	3.8000	0.8454	0.1396	0.7287
Comparison	0.7198	3.8947	0.8408	0.0537	0.6691
Difference	7.769%	-2.432%	0.548%	159.862%	8.917%

Table 1 shows a summary of all scores. Each row shows the average score across 20 evaluated papers, showing both absolute values for each method and the relative percentage difference. The "Difference" row highlights where the framework outperforms or underperforms compared to the baseline.

It is important to note that due to the scope of this project, 20 papers are used and evaluated, which is a limited sample size. Thus, results should be interpreted as indicative trends. A more extensive evaluation would be future work to confirm these findings statistically.

5.1 LLM-Based Evaluation (Gemini and Claude)

Using two independent LLM evaluators (Google's Gemini 1.5 Pro and Anthropic's Claude 3), ratings on four qualitative dimensions are obtained: factual accuracy, clarity, structural completeness, and keyword relevance. Gemini consistently favoured the framework across most criteria.

Table 2. Gemini 1.5 Pro evaluation comparison between framework and comparison abstracts

	Factual Accuracy	Clarity	Structural Completeness	Keyword Relevance	Average
Framework	4.3553	4.4342	3.5395	4.5789	4.2269
Comparison	4.0921	4.8421	3.4211	4.4605	4.2039
Difference	6.431%	-8.424%	3.462%	2.655%	0.548%

It rates the framework's abstracts 6.43% more factually accurate, 3.46% more structurally complete, and 2.66% more aligned with relevant keywords. The only drawback noted was a −8.42% drop in clarity compared to the baseline. On average, this had a .55% overall gain, suggesting a slight but positive improvement with the structured framework compared to the baseline.

Table 3. Claude evaluation comparison between framework and comparison abstracts

	Factual Accuracy	Clarity	Structural Completeness	Keyword Relevance	Average
Framework	4.2375	3.7500	3.0375	4.1750	3.8000
Comparison	4.0921	4.2368	3.1842	4.0658	3.8947
Difference	3.553%	-11.491%	-4.607%	2.686%	-2.432%

Claude, in contrast, presents a more critical evaluation. While it agreed that the framework produced abstracts with higher factual accuracy (+3.55%) and stronger keyword relevance(+2.69%), it rates the framework substantially lower in clarity (−11.49%) and structural completeness (−4.61%). These scores led to an overall average drop of −2.46% for the framework compared to the baseline. Detailed visualizations of these results, comparing both models and evaluators, can be found in Appendix D (Figure 2).

The justification outputs shows the reason for these scores. Claude's clarity results are often related to feedback such as "less concise and harder to follow" or "repetitive phrasing", particularly in the Introduction and Discussion sections. Likewise, its structural low scores frequently references "missing summarization of specific methodological components" or "lack of closure". By contrast, Gemini emphasises that the framework includes relevant elements in the sections, even noting that the Methods section is occasionally "technically overloaded" or "poorly organised", which likely contributed to its loss of clarity.

Justifications from both models support the improved ratings in factual accuracy, frequently claiming that the framework "accurately summarises the methods/results without hallucinating" and that "factual claims are well-supported by the source". The consistent inclusion of relevant domain terminology lead both models to acknowledge that the generated abstracts "effectively incorporated most keywords without disruption".

The differences between the two models suggest they give importance to slightly different aspects. Gemini appears to prioritise structure and conceptual alignment, which benefits the framework, while Claude appears to be more sensitive to clarity and surface-level cohesion, penalising the framework for less fluid writing. Despite this difference, both models agree on two key points: the framework improves factual grounding and keyword incorporation, and it may struggle with clarity under certain conditions. This reinforces the idea that while structured prompting enhances coverage and accuracy, it can introduce complexity that may reduce readability or narrative flow from an LLM's perspective.

5.2 Section-Wise LLM Ratings (IMRAD)

The evaluation scores by IMRAD section shows where the structured framework most improves abstract quality and where it presents problems. The assessments from both Claude and Gemini reveal areas of overlap and differences that highlight strengths and weaknesses.

According to Claude, the framework outperforms the baseline in the Methods (+3.51%) and Results (+4.06%) sections. However, Claude rates the framework lower in Introduction (−8.21%) and Discussion (−7.48%). These preferences are consistent with Claude's justifications, which repeatedly praised the baseline for being "fluent", "easier to read", especially in opening and closing sections.

In contrast, Gemini gives a more favourable view of the framework, especially in Results and Discussion. It rates the framework +8.97% higher in Results and +5.57% higher in Discussion, suggesting that it recognised better coverage on those areas.
The Introduction also received a higher score (+8.411%). However, Gemini showed a sharp drop in Methods performance (−14.65%), favouring the baseline by a large margin. According to Gemini's justifications, this penalty often stemmed from complaints about "overly detailed or fragmented phrasing" that "reduced readability" in technical sections, highlighting that even structurally correct content can have a drop in score if it overwhelms the reader.

Together, these trends show where each model saw value:

- Both Claude and Gemini praise the framework's Results section, highlighting its capacity to reflect empirical findings more accurately and thoroughly.
- Claude alone favoured the framework's Methods section, likely due to its appreciation for the technical completeness,

Table 4. Claude evaluation scores comparing framework and comparison abstracts per IMRAD section

Section	Factual Accuracy	Clarity	Structural Completeness	Keyword Relevance	Average
Introduction					
Framework	4.400	3.850	3.150	4.350	3.938
Comparison	4.579	4.368	3.684	4.526	4.289
Difference					-8.21%
Methods					
Framework	4.000	3.750	2.900	3.950	3.650
Comparison	3.579	4.000	2.789	3.737	3.526
Difference					3.51%
Results					
Framework	4.300	3.800	3.150	4.250	3.875
Comparison	4.000	4.158	2.842	3.895	3.724
Difference					4.06%
Discussion					
Framework	4.250	3.600	2.950	4.150	3.738
Comparison	4.211	4.421	3.421	4.105	4.039
Difference					-7.48%

Table 5. Gemini evaluation scores comparing framework and comparison abstracts per IMRAD section

Section	Factual Accuracy	Clarity	Structural Completeness	Keyword Relevance	Average
Introduction					
Framework	4.737	4.737	3.895	4.947	4.579
Comparison	4.579	3.895	3.737	4.684	4.224
Difference					8.411%
Methods					
Framework	3.526	3.895	2.947	3.737	3.526
Comparison	3.947	4.789	3.368	4.421	4.132
Difference					-14.650%
Results					
Framework	4.526	4.474	3.579	4.684	4.316
Comparison	3.737	4.789	3.105	4.211	3.961
Difference					8.970%
Discussion					
Framework	4.632	4.632	3.737	4.947	4.487
Comparison	4.105	4.895	3.474	4.526	4.250
Difference					5.573%

while Gemini strongly preferred the baseline, due to its simpler expression and better organisation of content.
- In Introduction and Discussion, Gemini and Claude have almost opposite scores, suggesting that evaluators differ in how they judge clarity and interpretation.

Overall, this mixed model feedback highlights that the framework consistently delivers stronger factual coverage, but its performance in sections like the Introduction or Methods can vary depending on the evaluator. This suggests that future improvements should focus on balancing technical detail with clearer language, simplifying dense sections and improving flow, to better meet both structural and readability expectations.

5.3 Semantic Similarity via Facet Score

Table 6. Summary of the face metric scores, comparison between framework and comparison abstracts

	Introduction	Methods	Results	Discussion	Average
Framework	0.7993	0.7391	0.7206	0.8439	0.7757
Comparison	0.7834	0.6682	0.6658	0.7617	0.7198
Difference	2.036%	10.615%	8.225%	10.787%	7.773%

A semantic similarity at the facet level is calculated to quantify how closely each abstract section matches the content of the corresponding section of the original paper. The framework achieved a higher average semantic similarity in all sections. On average, the facet score (cosine similarity between the Sentence-BERT embeddings of the generated and original section) increased from 0.7198 with baseline to 0.7757 with the framework, a relative improvement of about 7.8%. The gains varied by section, ranging from roughly +2.0% to +10.8%. The smallest improvement was in the Introduction (+2.0%), where even the baseline abstracts covered the general topic reasonably well. In contrast, the Methods and Discussion sections saw much larger jumps (around +10.6% and +10.8% higher similarity, respectively). This implies that the baseline often omits or does not correctly summarise the methodological details and the discussion points of the papers. The structured framework was able to include those key details, making its Methods and Discussion sections much more aligned with the source text. The Results section also improved substantially (+8.2%). Detailed visualisations of these results can be found in Appendix D (Figure 3). These increases in semantic overlap suggest that the framework's abstracts are more faithful to the original content. The structured prompts and keyword feedback loop likely contributed to this alignment, ensuring that no key content "facet" was left out or hallucinated.

5.4 Readability

In addition to content accuracy, the readability of the generated abstracts was evaluated using the Flesch Reading Ease score. Here, higher scores indicate text that is easier to read. The structured framework's abstracts were found to be significantly more readable than the baseline's. On average, the framework abstracts scored 13.96 on the Flesch scale, compared to a remarkably low 5.37 for the baseline. For context, both numbers indicate a very dense, academic style (as expected for deep learning papers), but the baseline output is nearly at the floor of the readability scale. The framework's 13.96 score, while still indicating difficult text, is 160% higher than the baseline's score, reflecting a meaningful improvement in clarity of expression.

However, these readability scores may seem contradicting with the lower clarity ratings assigned by the LLM evaluators in earlier sections. This discrepancy can be explained by the different linguistic dimensions these metrics measure. Readability focuses on surface level features, such as sentence length and complexity, whereas LLM clarity scores evaluate cohesion, conciseness, and narrative organisation. In fact, both Claude and Gemini specifically noted that some framework sections, particularly Methods and Introduction, were "overly detailed" or "fragmented", which reduced their perceived clarity despite being easier to follow.

This reinforces the idea that improving linguistic accessibility does not ensure clarity measured by LLM evaluators. Thus, future refinements should aim to maintain the framework's improved readability while reducing redundancy and improving flow.

5.5 NLI Entailment

To evaluate factual consistency, a natural language inference (NLI) model is applied to each abstract, measuring how much of the generated content is entailed (supported) by the original paper. The framework shows a clear advantage, indicating fewer hallucinations and unsupported claims. Specifically, 72.87% of the statements in the framework's abstracts were classified as entailed by the source text, compared to 66.91% for the baseline. This is an 8.917% relative increase in entailment, meaning the structured abstracts contain a higher proportion of facts that can be verified by the original paper. The framework's iterative keyword-guided generation likely achieved a better score by continuously checking and enforcing the presence of relevant terms, thereby anchoring the content closer to the original. The result is that the framework abstracts are more trustworthy. Thus, they seem to cover more of the key points (as seen in the facet similarity scores) and also do so with statements that the source material can support. This higher entailment aligns with the LLM evaluators' judgment of better factual accuracy for the framework. Together, the evidence points to the framework effectively reducing hallucinations and increasing factual alignment compared to the unguided summary approach baseline.

5.6 Keyword Validation Loop

This section investigates whether the iterative keyword validation loop introduced in the framework contributes to the abstract quality. Specifically, the analysis examines how evaluation scores evolve across the five iterations, each triggered only if a previous attempt failed to include the required keywords. The goal is to determine if enforcing the inclusion of keywords over multiple prompts improves abstract quality.

Table 7 shows the number and percentage of best-scoring abstracts (those that achieved the highest evaluation score for each metric) that were generated in each of the five possible iterations. The results indicate that the majority of best abstracts were generated within the first two iterations.

Table 7. Number and percentage of best abstracts across multiple evaluation metrics over five iterations.

Iteration	Facet Score		Readability		Claude		Gemini		NLI Entailment	
	n°	%	n°	%	n°	%	n°	%	n°	%
1	8	47.06%	14	82.00%	7	44.00%	8	47.06%	7	41.18%
2	9	52.94%	3	18.00%	8	50.00%	7	41.18%	10	58.82%
3	0	0.00%	0	0.00%	0	0%	1	5.88%	0	0.00%
4	0	0.00%	0	0.00%	0	0%	1	5.88%	0	0.00%
5	0	0.00%	0	0.00%	1	6.00%	0	0.00%	0	0.00%

Iterations after the second rarely produced superior outputs: iterations 3, 4, and 5 together accounted for 3.57% of the best scores across all metrics. This suggests that in most cases, quality gains occur early within the first or second iteration after enforcing keyword coverage. While the first iteration often performs well, it is notable that a very relevant amount of best abstracts were generated during the second iteration, especially for metrics such as facet similarity, NLI entailment, and LLM judgments. This indicates that the framework's iterative loop is an effective enhancement mechanism. By forcing the model to regenerate content with increased attention to the missing keywords, the second iteration often succeeds in correcting incomplete or imprecise outputs that the first attempt missed.

Table 8. Summary statistics of improvement percentages across evaluation metrics

	Facet Score	Readability	Claude	Gemini	NLI Entailment
Average increase	5.27%	22.47%	3.42%	3.49%	12.11%
Max	13.85%	56.94%	6.98%	10.17%	40.00%
Min	0.19%	4.22%	1.82%	1.33%	1.35%

Table 8 summarises the relative percentage improvements in the evaluation metrics for cases where scores increased. On average, the iterative process has moderate but consistent gains across all evaluation metrics, with readability and factual consistency showing particularly strong improvements. Maximum improvements per paper were large, indicating that in specific instances, the validation loop led to important enhancements in abstract quality.

Overall, these findings suggest that the keyword validation loop is generally beneficial, particularly in the first two iterations. It enables recovery from initially incomplete outputs by enforcing the inclusion of key content elements. While increments tend to diminish after the second attempt, the second iteration alone is responsible for a considerable amount of the highest score outputs across most metrics. However, given the low utility of later iterations, future implementations could consider adding stopping mechanisms after two attempts to reduce unnecessary use of resources without sacrificing output quality.

6 CONCLUSION

The aim of this thesis is to investigate the extent to which structured, keyword-guided prompts can improve the factual accuracy, coverage, and relevance of LLM-generated abstracts for deep learning papers. To address this question, a generation framework is introduced that uses an IMRAD structure, incorporates author-provided keywords, and applies a feedback loop to iteratively refine the abstract.

With respect to SRQ1, the results show that abstracts generated using the proposed structured framework consistently outperform those produced with a flat, single-prompt baseline. Improvements are observed across both automated and LLM metrics, especially in factual consistency, and structural completeness. These findings demonstrate that guiding generation through explicit keyword anchoring significantly enhances abstract quality.

SRQ2 focuses on the role of the iterative keyword validation loop. Experiments show that the loop provides the greatest improvements within the first one or two iterations. However, further iterations showed small returns.

Future work could include exploring the applicability of this framework apart from deep learning papers, including fields such as physics, biology, or the social sciences. This may involve adapting keyword extraction methods and adjusting how the text is divided into IMRAD sections.

Involving human experts in the evaluation process could also strengthen the assessment of abstract quality. Expert feedback would be especially useful for identifying specific inaccuracies and evaluating clarity from a researcher's perspective.

Finally, the framework could be developed into a tool to support researchers in writing abstracts. It could allow users to edit, review, or approve each section during generation, combining automated help with human input.

In summary, the framework developed in this study demonstrates that structured keyword-guided prompting can substantially enhance the quality of LLM-generated abstracts in technical domains. While the approach is currently designed for deep learning papers, it offers a foundation for other applications in scientific summarisation.

REFERENCES

[1] [n. d.]. SciSummary: Use AI To Summarize Scientific Articles. https://scisummary.com/. Accessed: 2025-05-01.

[2] Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renée DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. 2024. Factuality Challenges in the Era of Large Language Models. *Nature Machine Intelligence* 6, 8 (2024), 852–863.

[3] Yizhe Chen et al. 2022. CTRLsum: Towards generic controllable text summarization. In *Proceedings of EMNLP 2022.*

[4] Felix Chern, Rishi Bommasani, Tianyi Zhang, Jianyu Zhang, et al. 2024. SCALEEVAL: A Scalable and Reliable Benchmark for LLM-as-a-Judge. arXiv preprint arXiv:2401.16788. https://arxiv.org/abs/2401.16788

[5] Comet Blog. 2024. LLM Juries: A New Approach to Fairer LLM Evaluation. https://www.comet.com/site/blog/llm-juries-for-evaluation/ Accessed: 2025-06-16.

[6] Murthy V Devarakonda, Gopal Krishnamoorthy, and Dina Demner-Fushman. 2017. Automated generation of structured biomedical abstracts: An exploratory study. *Journal of Biomedical Informatics* 72 (2017), 40–47. https://doi.org/10.1016/j.jbi.2017.03.001

[7] Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology* 32, 3 (1948), 221–233.
[8] Tanya Goyal and Greg Durrett. 2022. Evaluating Factuality in Generative Summarization with Dependency Graphs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2022)*. https://aclanthology.org/2022.naacl-main.345
[9] Tushar Goyal and Greg Durrett. 2022. News Summarization and Evaluation in the Era of GPT-3. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 4786–4811. https://doi.org/10.18653/v1/2022.acl-long.329
[10] James Hartley. 2004. Current findings from research on structured abstracts: an update. *Journal of the Medical Library Association* 92, 3 (2004), 368–371.
[11] Or Honovich, Thomas Scialom, Omer Levy, and Tal Schuster. 2022. True or False? Fake News Detection as Fact-Verification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*. https://aclanthology.org/2022.acl-long.431
[12] Zheheng Ji, Nayeon Lee, Rita Frieske, Tao Yu, Dan Su, Yan Xu, Qi Zhu, Xinyi Li, Wei Xu, et al. 2023. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys (CSUR)* (2023). https://arxiv.org/abs/2301.12017.
[13] Wonjoon Kim, Youngjoong Kim, Bill Yuchen Lin, and Xiang Ren. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models But Not Without Factual Backing. *arXiv preprint arXiv:2310.05657* (2023). https://arxiv.org/abs/2310.05657
[14] Mario Krenn, Luca Buffoni, Bruno Coutinho, and Michael Kopp. 2022. Predicting the Future of AI with AI: High-quality link prediction in an exponentially growing knowledge network. *Preprint* (2022). https://www.researchgate.net/publication/364126422 Accessed via ResearchGate.
[15] Han Chung Lee. 2024. LLM as a Judge: Challenges and Best Practices. https://leehanchung.github.io/blogs/2024/08/11/llm-as-a-judge Accessed: 2025-06-16.
[16] Haoran Li, Weiran Xu, Wenlin Yao, Zhiyuan Liu, and Tat-Seng Chua. 2020. Keywords-guided Abstractive Sentence Summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8199–8206.
[17] Junnan Li, Chengming Tan, Wayne Xin Zhao, and Ji-Rong Wen. 2018. Guiding Generation for Abstractive Text Summarization Based on Key Information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*. 2652–2661.
[18] Zhen Li et al. 2024. ISQA: Informative Factuality Feedback for Scientific Summarization. *arXiv preprint arXiv:2404.13246* (2024).
[19] Peter Liu, Behnam Ghavami, Kaiqiang Song, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Lost in the Middle: How Language Models Use Long Contexts. *arXiv preprint arXiv:2307.03172* (2023). https://arxiv.org/abs/2307.03172
[20] Xingyu Lu, Jingjing Liu, Shaohan Huang, and Xiaojun Wan. 2023. A Systematic Survey of Hallucination in Neural Summarization. arXiv preprint arXiv:2302.07459. https://arxiv.org/abs/2302.07459
[21] Aman Madaan et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *Advances in Neural Information Processing Systems (NeurIPS)* (2023).
[22] Ahalya Murugadoss, Nathan Lee, Byron Wallace, et al. 2025. Evaluating the Evaluator: Measuring LLMs' Adherence to Task Evaluation Instructions. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2025)*.
[23] Hyeonwoo Oh, Seungwoo Nam, and Yida Zhu. 2023. Structured abstract summarization of scientific articles: Summarization using full-text section information. *Journal of the Association for Information Science and Technology* 74, 2 (2023), 234–248.
[24] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM Evaluators Recognize and Favor Their Own Generations. arXiv:2404.13076 [cs.CL] https://arxiv.org/abs/2404.13076
[25] Pontus Plavén-Sigray, Granville J Matheson, Björn C Schiffler, and William H Thompson. 2017. Research: The readability of scientific texts is decreasing over time. *eLife* 6 (2017), e27725. https://doi.org/10.7554/eLife.27725
[26] Kevin Roose. 2023. Don't ban ChatGPT in schools. Teach with it. https://www.nytimes.com/2023/01/12/technology/chatgpt-schools-teachers.html. The New York Times.
[27] A Sezer, D Ayhan Başer, S Oztora, A Caylan, and H N Dağdeviren. 2022. The Importance of Keywords and References in a Scientific Manuscript. *Eurasian Journal of Family Medicine* 11, 4 (2022), 185–188.
[28] Luciana B Sollaci and Maria G Pereira. 2004. The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey. *Journal of the Medical Library Association* 92, 3 (2004), 364–371.
[29] Yixuan Song, Yuanhe Tian, Zihan Zeng, Xiaozhong Lin, Yanshan Xu, and Yulan He. 2024. FineSurE: Fine-grained Factuality Evaluation for Scientific Summarization. *arXiv preprint arXiv:2407.00908* (2024). https://arxiv.org/abs/2407.00908
[30] Yijia Wu, Yinghao Lin, et al. 2023. SummIt: Iterative Text Summarization via ChatGPT with Self-Evaluation. *Findings of the Association for Computational Linguistics: EMNLP 2023* (2023).
[31] Demian Ye, Andy Chen, Yuxiang Wang, Julian Tachella, Bill Yuchen Lin, and William Yang Wang. 2023. FLASK: Fine-grained Language Model Evaluation based on Alignment Skill Sets. *arXiv preprint arXiv:2307.10928* (2023).
[32] Yujia Zhang, Yixuan Shen, Esin Durmus, Claire Cardie, and Chenhao Tan. 2023. Benchmarking Large Language Models for News Summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 8448–8465. https://doi.org/10.18653/v1/2023.acl-long.470

AI USE STATEMENT

I used generative AI tools (specifically ChatGPT by OpenAI) to assist with grammar correction, language refinement, and improving the clarity of writing throughout this thesis. All ideas, analyses, and conclusions are my own.

A APPENDIX A

This appendix lists the regular expressions used to extract IMRAD sections and keywords.

IMRAD Section Patterns

All regex patterns are case-insensitive and match optional punctuation or numbering in headers:

- **Introduction:**
  `(#+\s*)?(introduction)[:\-]?\s*`
- **Methods or Materials and Methods:**
  `(+\s*)?(methods|materials and methods)[:\-]?\s*`
- **Results:**
  `(+\s*)?(results)[:\-]?\s*`
- **Discussion or Conclusions:**
  `(+\s*)?(discussion|conclusion[s]?)[:\-]?\s*`

These patterns also handle markdown headers (e.g., `Discussion`) and numbered formats like `3. Methods`.

Keyword Block Pattern

Keywords:
`(keywords|index terms)[:\-]?\s*(.*)`

Only the first match per document was used. Extracted strings were split into keywords using commas.

B APPENDIX B

This prompt was used for section-aware abstract generation:

> You are a scientific abstract generation engine. Your task is to generate a structured abstract in IMRAD format (Introduction, Methods, Results, Discussion) using the provided content from each section of the paper, which appears at the end of each corresponding instruction block in the format [SECTION NAME] section text. You must preserve factual accuracy and semantic alignment with the source content.
>
> ## FORMAT Output exactly these four sections: **Introduction** ... **Methods** ... **Results** ... **Discussion** ...
>
> ## GENERAL RULES • Do not fabricate, infer, or generalize beyond the source. • Use terminology, numerical results, and notation exactly as provided. • Avoid boilerplate; each sentence must reflect real content. • Do not copy verbatim — paraphrase while keeping facts intact.
>
> ## SECTION-SPECIFIC GUIDELINES
> **Introduction** - Clearly state the research question and motivation. - Frame the scientific challenge and briefly hint at the solution approach. - Optionally use domain-specific terms such as: {keyword_str}
> [INTRODUCTION SECTION] {intro}
>
> —
>
> **Methods** - Describe the study design, datasets, model architecture, features, tools, and training procedure. - Prioritise clarity and technical specificity (e.g., models used, metrics, hyperparameters). - List steps in order, mirroring the logical flow of the real Methods. - Optionally use domain-specific terms and relevant to this section such as: {keyword_str}
> [METHODS SECTION] {methods}
>
> — **Results** - Report all relevant numerical findings, comparisons, and performance metrics. - Prefer exact values over vague statements. - Preserve table results in natural language. - Optionally use domain-specific terms and relevant to this section, such as: {keyword_str}
> [RESULTS SECTION] {results}
>
> —
>
> **Discussion** - Emphasise key conclusions, implications, and any limitations noted. - Optionally incorporate broader ideas from: {keyword_str} - Do not speculate beyond the source content.
> [DISCUSSION SECTION] {discussion}

```
## FINAL INSTRUCTIONS Only return the abstract text. Do not include explanations, labels, or markdown
formatting beyond the section headers (**Introduction**, **Methods**, **Results**, **Discussion**).
```

Keyword Iteration Example

To enforce full keyword coverage, the system checks whether all required terms appear in the generated abstract. If not, it regenerates the abstract with a modified prompt. The process continues for up to 5 attempts.

Iteration Log Example

```
Attempt 1
Missing keywords: ['latent space', 'denoising']
Coverage: 60%

Attempt 2
Missing keywords: ['denoising']
Coverage: 80%

Attempt 3
Missing keywords: []
Coverage: 100%
```

Retry Prompt Format

When keywords are missing, the following message is prepended to the prompt:

```
The previous abstract was missing these important terms: [list of missing keywords].
Please regenerate the abstract using the same instructions, ensuring that all
missing terms appear in the correct sections without fabricating new information.
```

Example

For Attempt 2 in the example above (missing: `denoising`):

```
The previous abstract was missing these important terms: denoising.
Please regenerate the abstract using the same instructions, ensuring that
all missing terms appear in the correct sections without fabricating new information.
```

Output Structure of Abstracts

All generated abstracts follow a structured IMRAD format enforced by the prompt and post-processing. Each abstract contains four clearly labeled sections:

```
**Introduction**
A brief summary of the research context and motivation...

**Methods**
Details on the experimental setup, datasets, and model architecture...

**Results**
Key findings and performance metrics...

**Discussion**
Interpretation of results and conclusions drawn from the study...
```

C APPENDIX C

This is the simpler prompt used to generate comparison abstracts:

```
Generate a structured abstract using bolded section titles (Introduction, Methods, Results, Discussion) based on
this paper text: {paper_text}
```

Each section title must be surrounded by double asterisks. Do not include headers, quotes, or formatting beyond the abstract itself.

These simple prompt abstracts have the same structure described in Appendix B

D APPENDIX D

This is the prompt used for both Gemini and Claude evaluation.

You are an expert scientific writing evaluator. Your task is to assess the quality of a generated {facet} section (e.g., Introduction, Methods, Results, or Discussion) based on its alignment with the original paper.
You will evaluate the generated section on the following four criteria using a 1–5 integer scale, where 1 = poor, 3 = average, and 5 = excellent. Justify each score with 1–2 concise sentences.
Evaluation Criteria: 1. Factual Accuracy – Is the information factually correct and consistent with the original? Avoid hallucinations or incorrect claims. 2. Clarity – Is the section written clearly and coherently? Are the sentences well-formed and easy to understand? 3. Structural Completeness – Does the section include all essential elements expected in this part of a scientific abstract? 4. Keyword Relevance – Are key concepts and terminology from the original reflected in the generated section?
IMPORTANT: The response must be a valid JSON object. Do not include markdown, code blocks, or additional text. Return only the JSON object.
Format: { "factual_accuracy": <1-5>, "clarity": <1-5>, "structural_completeness": <1-5>, "keyword_relevance": <1-5>, "justification": "One or two sentences explaining your ratings." }
[Original {facet} Section]{source_text}
[Generated {facet} Section]{generated_text}

Output Structure of Evaluation

Each evaluation result is a valid JSON object:

```
{
  "factual_accuracy": 5,
  "clarity": 4,
  "structural_completeness": 5,
  "keyword_relevance": 5,
  "justification": "The generated section accurately summarises the content,
  includes key methods and terminology, and is clearly written."
}
```

The script saves these results per paper, mode (framework or comparison), and section (Introduction, Methods, Results, Discussion) into a final CSV file for analysis.

E APPENDIX E

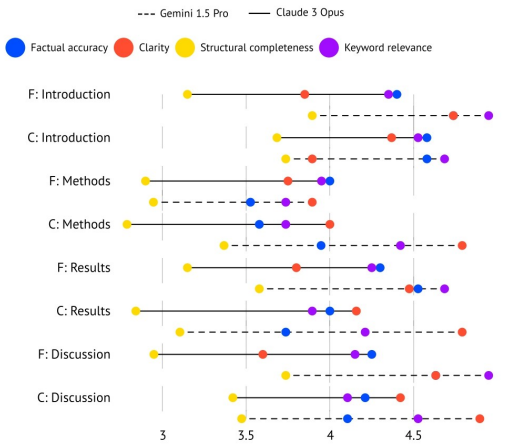

Fig. 2. Evaluation results of all generated abstracts with Claude 3 Opus and Gemini 1.5 Pro

Figure 2 compares the proposed framework (F:) and the baseline (C:) across all IMRAD sections. Each dot represents a score given by Claude 3 Opus or Gemini 1.5 Pro for one of four criteria: factual accuracy, clarity, structural completeness, and keyword relevance.

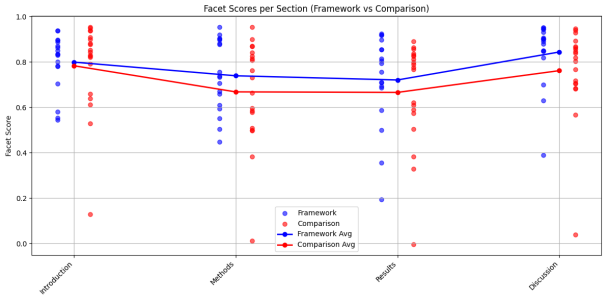

Fig. 3. Distribution of evaluation scores across facets and models

Figure 3 shows the distribution of evaluation scores for each IMRAD section, allowing visual comparison between the two generation strategies and the two evaluators.