Analysis and improvents of GLIMPSE - Pragmatically Informative Multi-Document Summarization for Scholarly Reviews

Stefano Brilli
Politecnico di Torino
Torino, Italy
s344207@studenti.polito.it

Stefano Gamba
Politecnico di Torino
Torino, Italy
s316675@studenti.polito.it

Pouria Mohammadalipourahar
Politecnico di Torino
Torino, Italy
s327015@studenti.polito.it

Valeria Tedesco
Politecnico di Torino
Torino, Italy
s340892@studenti.polito.it

***Abstract*—According to the original publication [1], scientific peer review is crucial to ensure the quality of academic publications, but the increasing number of paper submissions has strained the review process. Area chairs must analyze a growing volume of reviews to identify key arguments. This paper introduces GLIMPSE, a summarization method that provides a concise yet comprehensive overview of scholarly reviews. Unlike consensus-based approaches, GLIMPSE extracts both common and unique opinions, leveraging uniqueness scores based on the Rational Speech Act framework to identify relevant sentences. Experimental results demonstrate that GLIMPSE produces more discriminative summaries than baseline methods according to human evaluation while maintaining comparable performance in automatic metrics.**

In this text, we will try to improve the original GLIMPSE framework by implementing several extensions. The aim of the work is to improve the completeness and adaptability of the proposed GLIMPSE methodology. As a first approach, we will try to use a hybrid methodology to generate summaries, using both an extractive and an abstractive approach. We will then move on to using other summary models and will try to extend the domain by introducing the management of reviews written in languages other than English. We will then try to introduce an adaptive RSA reclassification process, which integrates semantic evaluation metrics. Finally, we will use a Large Language Model (LLM) as an expert to try to evaluate the quality of the generated summaries.

I. Introduction

GLIMPSE proposes a new methodology for summarizing scholarly opinions by overcoming the shortcomings of previous methods which did not account for the shared and divergent views in reviews. Indeed, this approach enables the synthesis of all aspects of the divergent reviews as it takes into account both the concordant and discordant opinions. GLIMPSE incorporates a new scoring technique that works with Rational Speech Act (RSA) framework that conceptualizes communication in a collaborative speaker-listener context as a 'reference game'. Each review is treated as an object and the summary is supposed to be a unique representation of the review object. Each potential summary provided by the RSA framework is assigned a score that reflects the degree of uniqueness of the summary and with that, it becomes possible for GLIMPSE to retrieve both the consensus and dissenting views found in the reviews. This gives rise to more detailed summaries that portray the many-sided nature of scholarly peer review. This paper expands on GLIMPSE and proposes new research on further improving the summarization of scholarly reviews.

II. Preparation of the repository

The repository can be found on GitHub. The code can be run using the Jupyter Notebook provided in the repository.

III. Extensions

In order to improve the original work, we implemented some extensions to the code. In particular, we focused on

- adding new evaluation methods to the original one, to show different ways of evaluating the results
- augmenting the original datasets by adding new records
- trying to improve the framework on multi-language documents

1) **Hybrid Summarization Extension: Integrating Extractive and Abstractive Techniques**
In this extension, our framework first applies extractive methods to identify and preserve the most critical information from each scholarly review by segmenting the text into sentences and selecting the three most salient ones. These sentences, which capture the essential content and context of the review, are then used as input for a transformer-based abstractive summarization model (e.g., `facebook/bart-large-cnn` [4]). Configured with a top-p sampling strategy (with parameters such as `max_new_tokens = 200`, `top_p = 0.95`, and `num_return_sequences = 8`), the model generates multiple candidate summaries. This hybrid approach capitalizes on the strengths of both extractive and abstractive techniques—preserving the original content while enhancing fluency and coherence—to produce summaries that are both succinct and informative.

2) **Aspect-Based Summarization Extension: Integrating Topic Modeling with RSA Decoding**
This extension introduces a thematic dimension to the summarization process by employing latent topic modeling techniques, such as BERTopic [2], to extract dominant topics from a corpus of scholarly reviews. Each review is assigned an aspect label based on its dominant topic, and an aspect prompt is generated by extracting the top keywords associated with that topic. This prompt is then concatenated with the original review text to create an enriched input that highlights thematic nuances. A transformer-based summarization model processes this augmented input using RSA contextual decoding to generate multiple candidate summaries. An RSA-based re-ranking module then computes a likelihood matrix between the source text and each candidate, iteratively refining the consensuality scores until full convergence is achieved for each paper. The candidate with the highest refined RSA score (denoted as `best_rsa`) is selected, ensuring that the final summary is not only coherent but also closely aligned with the specific thematic content of the review.

3) **Multi-Model RSA Re-Ranking and Evaluation Extension**
To further enhance diversity and robustness in candidate generation, this extension leverages multiple pre-trained summarization models, such as `facebook/bart-large-cnn` [4], `sshleifer/distilbart-cnn-12-6` [5], and `Falconsai/text_summarization` [6]. Each model produces its own set of candidate summaries, which are then combined into a unified candidate pool. An RSA-based re-ranking process computes a likelihood matrix for each candidate against the source text, with iterative log-softmax updates producing refined speaker scores and uniqueness scores. The best candidate summary is selected based on these refined scores. In addition, the quality of the generated summaries is assessed using evaluation metrics like ROUGE and BERTScore [3]; experimental results indicate a mean BERTScore F1 of approximately 0.8453, confirming a high degree of semantic overlap with gold-standard meta-reviews. All these outputs, along with detailed scoring metrics and the identity of the best-performing model, are compiled into a CSV file for comprehensive analysis.

4) **Multi-language extension**
This extension aims to analyze the behavior of the framework when the input document contains reviews written in languages other than English. The underlying rationale is that, in the hypothetical future, scientific papers may be reviewed in multiple languages simultaneously.
As a starting assumption, we consider the following scenarios:

- A paper written in English leads to a review written in English.
- A paper written in Spanish leads to a review written in Spanish.

A. Methodology

The steps to implement in this extension are:

- Creating an input document containing both records written in English and in another language. Specifically, we selected Spanish as the target language and employed the `Helsinki-NLP/opus-mt-en-es` model [7] to translate the original *all_reviews_2017.csv* file into Spanish. As a result, we obtained a dataset containing 3,025 records available in both English and Spanish.
- In order to improve the abstractive part of the summary, we looked for a suitable model to generate summaries both in English and in Spanish. Because of limited resources, we chose to fine-tune the `mt5-small-finetuned-amazon-en-es` model [8] for our purposes.
- Splitting the input dataset into three subsets:
  - A training set: 80% of the input records
  - The remaining records are further subdivided in validation set (80%) and test set (20%)

B. Fine-Tuning Approach

First of all, we generate the scores of the original model in order to know how the model is able to generate summaries of the reviews. Then, we prepare the fine-tuning part by setting the pipeline. Since the generative model was already trained on text written both in English and Spanish, it makes no sense to fine-tune the entire network. So, the idea is to implement a transfer learning approach and freezing the early layers of the model, since modifying them could lead the loss of the network to diverge. We decided to fine-tune only the last decoder block (*decoder.block.7*), the final normalization layer (*final_layer_norm*) and the generative layer (*lm_head*). With this setting, we got **131205120/300176768 (43.71%)** trainable parameters. In the following, the hyper-parameters we employed for fine-tuning the model.

- *Embedding size of the input records*: **512**
- *Embedding size of the input records*: **100**
- *learning rate*: **5e-5**
- *optimizer*: **Adam**
- *training batch size*: **16**
- *validation batch size*: **8**
- *weight decay*: **0.001**
- *number of epochs*: **5**
- *metric for the best model*: **RougeL**

C. Results and Observations

At the end of the process, we did not get valuable results. In particular, we noticed that the training loss did not

change during the fine-tuning process. This could be due to the fact that network overfitted the data, since the already trained model already knew simple sentences written in English and Spanish.

D. Possible Improvements

Several strategies could be employed to enhance the model and, consequently, improve the final results. These include:

- **Using a larger dataset**: Increasing the amount of training data could help the model generalize better and reduce overfitting.
- **Starting with a more powerful pre-trained model**: Utilizing a stronger baseline model and applying light fine-tuning could yield better performance.
- **Applying data augmentation techniques**: Generating variations of the existing data could enrich the training set and improve the model's robustness.
- **Implementing alternative evaluation methods during training**: Exploring different evaluation metrics or strategies could provide deeper insights into the model's performance and guide the fine-tuning process more effectively.

5) **Adaptive RSA Re-Ranking Extension: Iterative Re-Ranking with Combined Evaluation**
This extension refines the re-ranking process by introducing an adaptive, iterative mechanism that integrates both RSA probabilities and semantic evaluation metrics. Candidate summaries generated by a pre-trained model are initially scored by computing a likelihood matrix between the source text and each candidate. The RSA distributions are then initialized in log-space and iteratively updated using log-softmax operations until convergence is reached (typically within two iterations, as demonstrated by the mean absolute difference falling below a preset threshold). The best candidate summary is first selected based solely on the final RSA score (`best_rsa`). In parallel, BERTScore F1 values are computed for each candidate, and a combined score is derived by blending the RSA probabilities with the BERTScore [3] using a weighting parameter α. The candidate with the highest combined score (denoted as `best_combined`) is chosen as the final output. Overall, this adaptive re-ranking process has been shown to yield an overall mean BERTScore F1 of approximately 0.87, indicating that the approach effectively produces summaries with strong semantic alignment to the reference outputs.

E. Further evaluation methods

In the final paragraph of the article [1], the authors suggest that a natural follow-up to their work would be a human evaluation of the summaries generated by their model. Such an evaluation would require the participation of experts in the field relevant to the reviewed scientific article, ensuring that they can assess whether the summary adequately captures the unique and common ideas present in all the reviews of the article. Due to the unavailability of such costly resources, we opted to emulate these experts by leveraging a Large Language Model (LLM) as a surrogate expert. To maximize the reliability of the LLM used as an expert, we adopted several strategies outlined in [10].

Model Selection

The LLM employed is GPT-4o-mini. This choice is motivated by the fact that, being a model trained through Reinforcement Learning from Human Feedback (RLHF), it is inherently aligned with human values and reasoning, making it suitable for expert-level evaluations.

In-Context Learning Methods

To prepare the selected LLM for the evaluation task, we designed three input prompts. Each prompt contains a definition of the evaluation task to be performed, a Chain of Thought that the LLM should follow during execution, and a predefined JSON output format that the LLM must adhere to when providing its evaluation. We deliberately avoided including few-shot examples, as their usefulness would have required expert human involvement in their construction. Poorly designed examples, on the other hand, could have compromised the reliability of the LLM-as-expert approach.

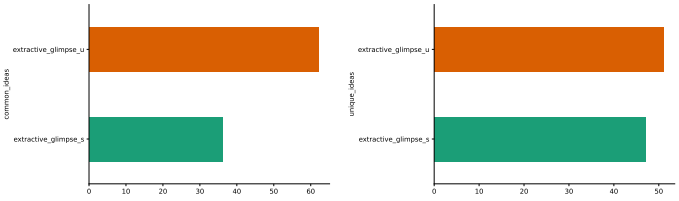
The first prompt instructs the LLM to conduct a pairwise comparison. Given a list of reviews and two summaries generated from this list, the LLM must determine which of the two summaries better captures the common and unique ideas present in the reviews. This evaluation setup aligns most closely with human assessment, as indicated in [10]. The second prompt provides the LLM with a list of reviews and a single summary generated from that list, requiring the LLM to assign a score between 0 and 1 for the summary's ability to capture both common and unique ideas. The third prompt is similar to the second but asks the LLM to assign scores based on each question from the Seahorse evaluation framework [9], which was used in [1]. This approach was chosen to visualize the differences between these two evaluation systems. In all three prompts, both the list of reviews and the summary pairs for the pairwise comparison are randomly ordered to mitigate position bias effects, as suggested in [10].

Post-Processing Methods

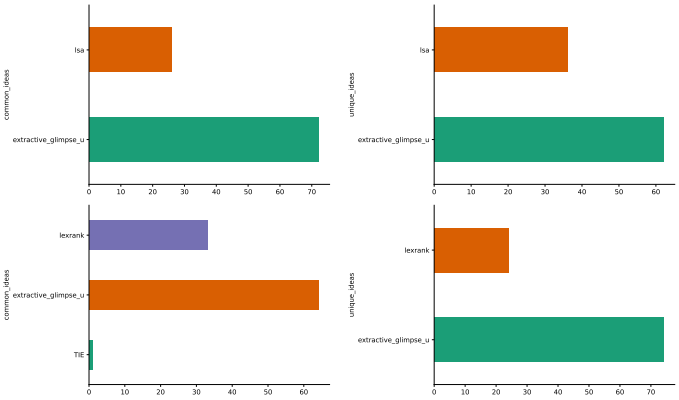
To ensure that the LLM outputs structured and computable results, we enforced strict adherence to predefined JSON formats within the prompts. Additionally, to stabilize the pairwise comparison results, we implemented multiple evaluation iterations for each summary pair, allowing us to determine the majority judgment or, in cases of ties, record an equal preference between the two summaries.

Results

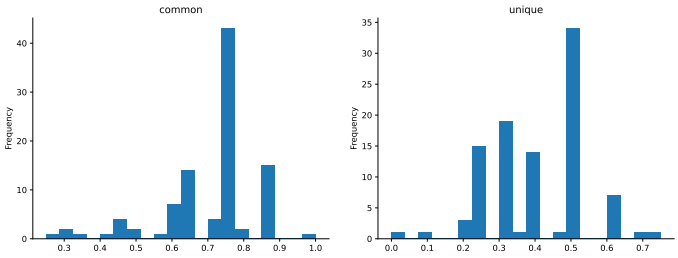
To generate results, we utilized a sample of the dataset from [1], constrained by limited hardware resources and time. Given that [1] indicates a preference for extractive summarization methods, we focused our analysis accordingly. After generating summaries using both the GLIMPSE-speaker and GLIMPSE-unique models, we conducted a pairwise comparison between these two approaches using the first prompt. The LLM determined that GLIMPSE-unique performed better in summarizing both common and unique ideas for the majority of the cases:



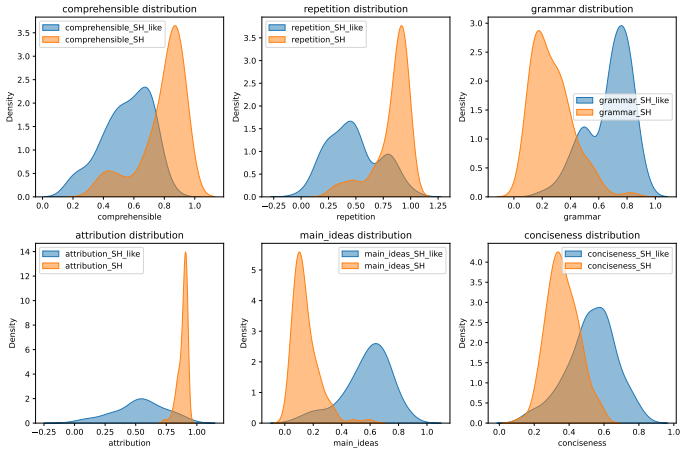
Subsequently, we conducted pairwise comparisons between this model and two baseline models used in [1], namely LSA and Lexrank. In both cases, the LLM judged GLIMPSE-unique superior in capturing both common and unique ideas in the majority of the cases:



Following this, we assigned scores to the GLIMPSE-unique model for its ability to summarize common and unique ideas using the second prompt. Here it is the score distribution:



Finally, we evaluated GLIMPSE-unique using Seahorse-like metrics and analyzed the deviation between its score distribution and that of the original Seahorse model:



REFERENCES

[1] Maxime Darrin and Ines Arous and Pablo Piantanida and Jackie CK Cheung, "GLIMPSE: Pragmatically Informative Multi-Document Summarization for Scholarly Reviews", 2024

[2] Grootendorst Maarten, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure", 2022

[3] Tianyi Zhang and Varsha Kishore and Felix Wu and Kilian Q. Weinberger and Yoav Artzi, "BERTScore: Evaluating Text Generation with BERT", 2020

[4] Mike Lewis and Yinhan Liu and Naman Goyal and Marjan Ghazvininejad and Abdelrahman Mohamed and Omer Levy and Ves Stoyanov and Luke Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", 2019

[5] https://huggingface.co/sshleifer/distilbart-cnn-12-6

[6] https://huggingface.co/Falconsai/text_summarization

[7] https://huggingface.co/Helsinki-NLP/opus-mt-en-es

[8] https://huggingface.co/Natet/mt5-small-finetuned-amazon-en-es

[9] Elizabeth Clark1 and Shruti Rijhwani1 and Sebastian Gehrmann2 and Joshua Maynez1 and Roee Aharoni2 and Vitaly Nikolaev1 and Thibault Sellam1 and Aditya Siddhant1 and Dipanjan Das1 and Ankur P. Parikh1, "SEAHORSE: A Multilingual, Multifaceted Dataset for Summarization Evaluation", 2023

[10] Jiawei Gu and Xuhui Jiang and Zhichao Shi and Hexiang Tan and Xuehao Zhai and Chengjin Xu and Wei Li and Yinghan Shen and Shengjie Ma and Honghao Liu and Yuanzhuo Wang and Jian Guo, "A Survey on LLM-as-a-Judge", 2025