Self-Feedback Pipeline for Text Simplification using Large Language Models

Ansh Kumar Dev

1 Overview

Understanding complex language in research papers can be challenging. Large language models (LLMs) can help simplify these texts, but they may sometimes lose important information or the intended meaning. The project aims to simplify text using Large Language Models (LLMs) while preserving its original meaning. This simplification process utilizes a self-feedback pipeline, which refers to the iterative use of feedback generated by the model itself to simplify the text and retain the actual meaning. The outputs produced over multiple iterations are evaluated using BERT scores, which measure how similar the simplified text is in meaning to the original text.

2 Methodology

2.1 Self-feedback pipeline

The project uses a feedback pipeline that consists of three main components: initial text, Feedback, and Simplified text. The initial text here is the text that needs to be simplified for better understanding. The feedback component evaluates whether the simplification retains essential information and offers specific suggestions to enhance clarity and readability while preserving the original meaning.

The feedback pipeline works by first simplifying the initial text. Then, it gets the feedback on the simplified text and checks how well it is simplified and how much the actual meaning is retained. Then, the feedback is passed back into the algorithm with the initial text to further simplify the text, but considering the feedback this time. This process can be repeated iteratively until satisfactory results are obtained.

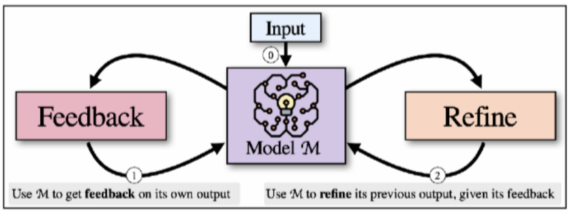
Input
0
Feedback
Model $\mathcal{M}$
Refine
1
2
Use $\mathcal{M}$ to get **feedback** on its own output
Use $\mathcal{M}$ to **refine** its previous output, given its feedback

Figure 1: Self Refine components and process. [1]

2.2 Models used

Models used for this task include: GPT-4o-mini, GPT-4o, GPT-4, GPT-4-turbo, and GPT-3.5-turbo.

These models have been specifically selected as some of the best offered by OpenAI. The purpose of using various models is to compare the performance of this self-feedback pipeline. The main goal of this analysis is to evaluate how well the pipeline performs across these top models.

Table 1: Parameter Settings for Simplification Process

Parameter	**Value**
Similarity Threshold	0.97
Maximum Iterations	3
Temperature	0.5

2.3 Evaluation metric

The BERT scores were utilized to assess various LLM models for text similarity. These scores indicate how closely the simplified texts resemble the original ones. A lower score signifies that the text is less similar to the original, while a higher score indicates a greater similarity between the two texts.

3 Experiments

The self-refine pipeline has been tested on five different models using five distinct prompts (original texts). The performance of each prompt across various models is visualized in Figure 2. All the prompts used for these experiments can be found in the data file provided with the code.

Additionally, the project code includes an experiment file that records each iteration, capturing both the original text and the simplified text generated after each iteration. This file also stores the corresponding BERT scores for every simplified text, indicating the similarity scores compared to the original text.

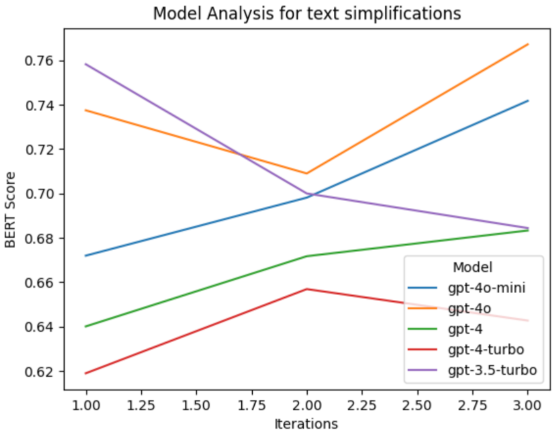

Figure 2: BERT Scores for 3 feedback iterations

4 Analysis

The text generated by the self-feedback pipeline has been significantly simplified over multiple iterations. Across the five different models, there is considerable variability in the BERT scores. The main objective of the analysis was to determine whether the text could be further simplified without losing its meaning. The analysis for a specific prompt is visualized in Fig. 2. Here, the large language models ’gpt-4o-mini’ and ’gpt-4’ performed quite well. As shown in the graph, the BERT scores for the outputs of these models (simplified text) increased with each iteration, indicating that they were effectively regaining the original meaning of the input text.

In this process, feedback helps the models understand the loss of information and allows the large language models (LLMs) to approximate the actual meaning of the text while making it simpler. Other models, such as ’gpt-4o’ and ’gpt-3.5-turbo,’ also performed well, although there was a significant loss of information during the second iteration. However, these models showed strong performance in the initial iterations, demonstrating their capability for the task. As the iterations progressed, they were able to regain information and achieved meanings that were more comparable to those generated by the other models.

These analyses were conducted across different prompts, and performance varied from text to text and from model to model. The similarity scores can serve as a benchmark for determining when to stop iterations to achieve better results.

5 Results

5.1 Model Performance Analysis

The evaluation of the self-feedback pipeline revealed distinct patterns among the models across three iterations. BERT scores indicated that GPT-4o and GPT-4o-mini initially had higher scores (approximately 0.74) compared to GPT-3.5-turbo, which had a score around 0.62. Most models showed improvement with feedback, generally reaching their peak scores by the second iteration.

GPT-4o consistently outperformed the others, achieving a high BERT score and demonstrating steady improvement across the iterations. GPT-4o-mini also performed well, exhibiting minor fluctuations but maintaining competitive scores. Notably, GPT-3.5-turbo experienced the most significant improvement of all the models.

5.2 Feedback Mechanism Effectiveness

The feedback mechanism helped all models recover information effectively, with the best results usually reached by the second iteration. Additional iterations provided minimal improvements, indicating a practical limit. However, results vary between texts, so the number of iterations may need to be adjusted depending on the content.

5.3 Cross-Model Analysis

The more advanced models performed consistently well throughout the process. The basic models, while starting with lower scores, improved more when given feedback. This means users can choose between models that work well immediately or models that improve significantly with practice.

6 Reference

[1] Aman Madaan, Niket Tandon, Prakhar Gupta, et al. *Iterative Refinement with Self-Feedback*. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*. Available online: Conference Paper