BeeManc at the PLABA Track of TAC-2024: RoBERTa for Task 1 – LLaMA3.1 and GPT-4o for Task 2

Zhidong Ling [1] **Zihao Li** [2] **Pablo Romeo** [3]
Lifeng Han [2,4*] **Goran Nenadic** [2]
[1] Tokyo Metropolitan University, [2] The University of Manchester
[3] Manchester Metropolitan University, [4] Leiden University
ling-zhidong@ed.tmu.ac.jp, 22466459@stu.mmu.ac.uk
{zihao.li-2,g.nenadic}@manchester.ac.uk, l.han@lumc.nl
* corresponding author

Abstract

This report is the system description of the BeeManc team for shared task Plain Language Adaptation of Biomedical Abstracts (PLABA) 2024. This report contains two sections corresponding to the two subtasks in PLABA 2024. In task one, we applied fine-tuned ReBERTa-Base models to identify and classify the difficult terms, jargon and acronyms in the biomedical abstracts and reported the F1 score. Due to time constraints, we didn't finish the replacement task. In task two, we leveraged Llamma3.1-70B-Instruct and GPT-4o with the one-shot prompts to complete the abstract adaptation and reported the scores in BLEU, SARI, BERTScore, LENS, and SALSA. From the official Evaluation from PLABA-2024 on Task 1A and 1B, our **much smaller fine-tuned RoBERTa-Base** model ranked 3rd and 2nd respectively on the two subtasks, and the **1st on averaged F1 scores across the two tasks** from 9 evaluated systems. Our LLaMA-3.1-70B-instructed model achieved the **highest Completeness** score for Task-2. We share our source codes, fine-tuned models, and related resources at `https://github.com/HECTA-UoM/PLABA2024`

1 Background

Health literacy, or the ability of individuals to comprehend and apply health information for informed decision-making, is one of the central focuses of the Healthy People 2030 framework in the US. Even though biomedical information is highly accessible online, patients and caregivers often struggle with language barriers, even when the content is presented in their native language.

The shared task PLABA aims to harness advances in deep learning to empower the automatic simplification of complex scientific texts into language that is more understandable for patients and caregivers. Despite substantial obstacles to effective implementation, the goal of the PLABA track is to improve health literacy by translating biomedical abstracts into plain language, making them more accessible and understandable to the general public [1]. Following our participation on the PLABA 2023 shared task using large language models (LLMs) such as ChatGPT, BioGPT, and Flan-T5, and Control Mechanisms (Li et al., 2024), in this work, we introduce our system participation to the PLABA 2024. Instead of end-to-end biomedical abstract simplification as in PLABA-2023, in this year, PLABA-2024 introduced more granular-steps, including Term Replacement for Task-1 and Complete Abstract Adaption for Task-2, which we will describe in detail for our methodologies via fine-tuning RoBERTa-Base model for Task-1 and prompting LLMs (LLaMa-3.1-70B and GPT4o) for Task-2.

2 PLABA 2024 Task 1: Term Replacement

2.1 Introduction for Task 1

Task 1 does not require a full adaptation process. Instead, the system will identify challenging terminology, determine appropriate strategies for addressing these terms, and offer suitable replacements. This task is split into three subtasks:

1A: Identifying non-consumer terms This is a Name Entity Recognition (NER) task. The objective is to to extract a list of exact phrases from the given abstracts, each representing a concept that a consumer would not understand.

[1] `https://bionlp.nlm.nih.gov/plaba2024/`

1B: Classifying replacement This is a multi-class multi-label token classification task. Systems should identify any and all types of replacements that would be appropriate for each identified non-consumer term.

1C: Generation Provide text for each positive label from 1B (except for the "OMIT" label). For this task, there aren't popular submissions, so we will skip this task.

2.2 Methodology

2.2.1 Corpus

In this task, we leveraged the PLABA corpus (Attal et al., 2022), which contains biomedical queries and the abstracts of top-ten papers corresponding to the query from PubMed. The sentences in each abstract are manually aligned and simplified by experts. In addition, the difficult phrases are tagged with proper labels indicating the replacement method. The original dataset is in JSON format and one sample is shown in Fig2.

There are five different kinds of replacement labels, and the task requires systems to assign these labels non-exclusively to the phrases.

- Substituted: the term is jargon with a common alternative (e.g. "myocardial infarction" can be "heart attack").
- Explained: there is no alternative, or the term is important to the topic, and it should be explained. For example: "This study looked at treatments for sleep apnoea (when you stop breathing while sleeping)."
- Generalized: the term can be replaced with a more general category without losing its significance. For example: "Clearing of the infection is confirmed with a [*Nucleic Acid Amplification Test (NAAT)*] common lab test."
- Exemplified: the term has a specific example that would give a general audience an idea of what it is. For example: "Depression is common in people with neurodegenerative diseases (like Parkinson's)."
- Omitted: The term is not relevant to understanding the sentence or too technical to explain, and does not need to appear in a consumer version.

To adapt this data to a form that is suitable for an NER task, we need to map these labels to the phrases in the sentences.

2.2.2 Data Preprocessing

We introduce how we modify the training data to a format for our task setting. For each sentence, terms that are difficult to read are given human-annotated replacement labels. Terms can be a single word or a phrase, and label types can also be multiple.

For task 1A, we define it as a word-level binary NER task. Because this subtask is covered by subtask 1B, we did not develop a system to specifically address this task. The output for task 1A is the words that are assigned not the O label in task 1B.

For Task 1B, labels must be mapped to subword-level labels during model fine-tuning, as the model's tokenizer segments sentences into subword units. The original training data shown in Figure 2 contains phrase-level annotations, which are not directly compatible with the subword-level classification required by our NER model. To address this issue, we implemented a two-step label transformation process. First, we decomposed the annotated phrases into individual words, applying the BIO tagging scheme: the first word of each phrase receives a B-[tag] (Beginning), while subsequent words receive I-[tag] (Inside) labels. Second, to accommodate the model's tokenizer, which further segments words into subwords, we developed the following labelling strategy:

- For words originally labeled with B-[tag]: the first subword maintains the B-[tag], while subsequent subwords receive I-[tag] labels
- For words originally labeled with I-[tag]: all resulting subwords inherit the I-[tag] label
- Context words outside the annotated phrases are labelled with *O* (Outside)

This approach ensures consistent label propagation from phrase-level annotations to subword-level classifications needed to train the NER model.

The classification uses the following label types:

- SUBSTITUTE
- EXPLAIN

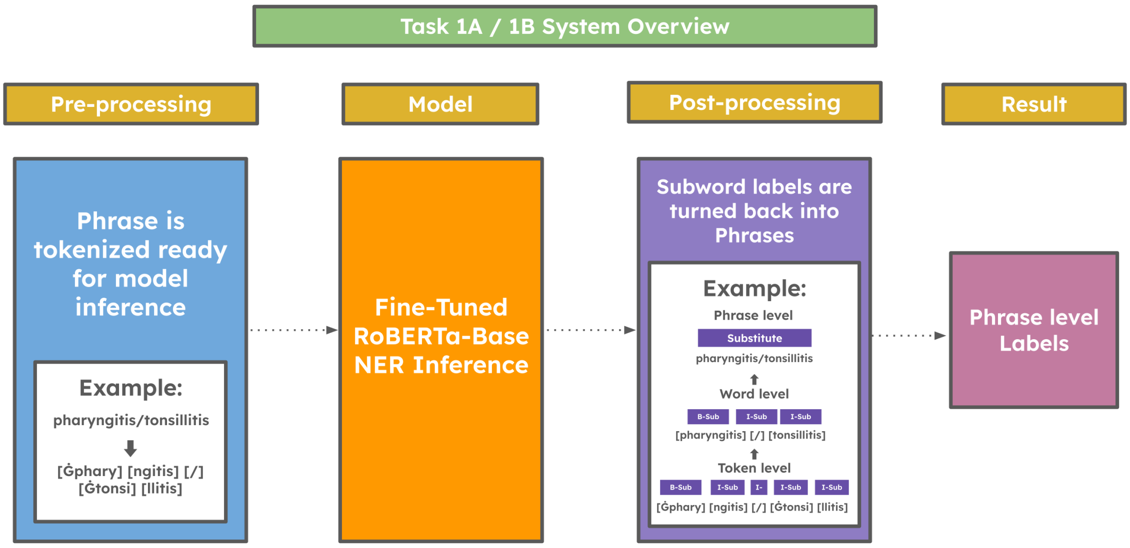

Figure 1: System overview for Task 1A and 1B

- EXEMPLIFY
- GENERALIZE
- OMIT

Each type follows the BIO tagging scheme with Beginning (B-) and Inside (I-) variants, plus an additional Outside (O) label, yielding eleven labels in total. Finally, we split the data into training, validation, and test sets with a ratio of 8:1:1, resulting in 836 sentences for training, 105 for validation, and 105 for testing. Table 1 shows the distribution of the label for each set.

Label	Train	Valid	Test
O	17703	2005	2411
B-SUBSTITUTE	1989	264	239
I-SUBSTITUTE	1385	181	191
B-EXPLAIN	1047	160	122
I-EXPLAIN	549	68	62
B-GENERALIZE	317	50	39
I-GENERALIZE	419	86	52
B-EXEMPLIFY	44	6	6
I-EXEMPLIFY	37	2	7
B-OMIT	309	45	50
I-OMIT	314	38	79

Table 1: Label Counts for Training, Validation, and Test Sets for Task 1B: 11 labels


```
{
    "Q1_A3": {
        "foveal regeneration": [
            [
                "SUBSTITUTE",
                "how well the fovea repaired itself"
            ],
            [
                "SUBSTITUTE",
                "healing of the center of the field of vision"
            ],
            [
                "EXPLAIN",
                "healing of the center of the field of vision"
            ],
            [
                "EXPLAIN",
                "(a part of the retina) regeneration"
            ]
        ],
        "visual acuity": [
            [
                "SUBSTITUTE",
                "vision"
            ],
```


Figure 2: Example of training corpus for Task 1

2.2.3 Model Description

For this multi-label, multi-class classification task, we selected the Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu et al., 2019) as our model of choice. RoBERTa is a transformer-based language model. It enhances the pretraining methodology of Bidirectional Encoder Representations from Transformers (BERT) to achieve superior performance in natural language processing tasks. Specifically, RoBERTa removes the next-sentence prediction objective, increases the amount of training data, and employs larger batch sizes and learning rates during training. These optimizations enable RoBERTa to outperform BERT on several benchmarks, demonstrating its advanced capabilities in understanding and generating human language. We used the open-source code from INSIGHTBuddy-AI for RoBERTa fine-tuning (Romero et al., 2024).

2.3 Experiments

Fine tuning process We add a linear layer after RoBERTa's final hidden layer. The system converts the output for each token into a set of probabilities for the 11 classes using the sigmoid function. We use Binary Cross Entropy as our loss function, specifically implementing it through PyTorch's BCEWithLogitsLoss[2]. A threshold of 0.5 is applied after the sigmoid function to determine if a token belongs to a particular label. Because we use sigmoid rather than softmax, each token can be assigned multiple labels. The detailed fine-tuning settings are shown in Table 7.

Post-processing After the system assigns subword labels, we merge them into word-level labels. Our strategy for combining subwords into words follows these rules:

1. A subword token beginning with the special character "Ġ" indicates the start of a new word.
2. If the subsequent token also begins with "Ġ", the current token is treated as a complete word and retains its label.
3. When a token with a leading "Ġ" is followed by one or more tokens without "Ġ", these tokens are merged into a single word. The merged word inherits the label from the initial "Ġ" token. For example, the tokens ['Ġdoc', 'us', 'ate'] with labels ['B-SUBSTITUTE', 'B-EXPLAIN'], ['I-SUBSTITUTE', 'I-EXPLAIN'], ['I-SUBSTITUTE', 'I-EXPLAIN'] merge to form 'docusate' with the labels ['B-SUBSTITUTE', 'B-EXPLAIN'].
4. If a word's first token is labelled 'O' but its later subwords have other labels, the whole word takes the first non-'O' label. For example: ['Ġdoc'('O'), 'us'('O'), 'ate'('I-SUBSTITUTE')] becomes 'docusate'('I-SUBSTITUTE').
5. Special sequence tokens (<s >, </s >) added by RoBERTa's tokenizer are removed as they are not part of the actual text content.

As the output of the format is phrase level, we need to merge the words back into phrases, here is our merging strategy for forming phrases:

1. When a token has a B-[tag], we combine it with all following I-[tag] tokens to form a phrase. For example: ['repeat'(B-SUBSTITUTE), 'infect'(I-SUBSTITUTE), 'ion'(I-SUBSTITUTE), 'rates'(I-SUBSTITUTE)] forms the phrase "repeat infection rates" with label SUBSTITUTE.
2. If we find an I-[tag] without a preceding B-[tag], we trace back to the first token starting with "Ġ" to mark the phrase beginning.

The output consists of identified difficult phrases and their corresponding replacement labels at the phrase level.

2.4 Results/Discussion

We report Automatic Evaluation of the word level evaluation for task 1B in table 2.

The RoBERTa-base model's performance shows significant limitations. While the word-level NER achieves a micro-F1 score of 0.836 when including ['O'] labels, this score drops substantially to 0.413 when excluding them. The confusion matrices for all 11 labels are shown in Fig 3. This performance degradation primarily stems from the model's poor prediction of low-frequency labels in the training data, particularly EXEMPLIFY, OMIT, and GENERALIZE. The

[2] `https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html#bcewithlogitsloss`

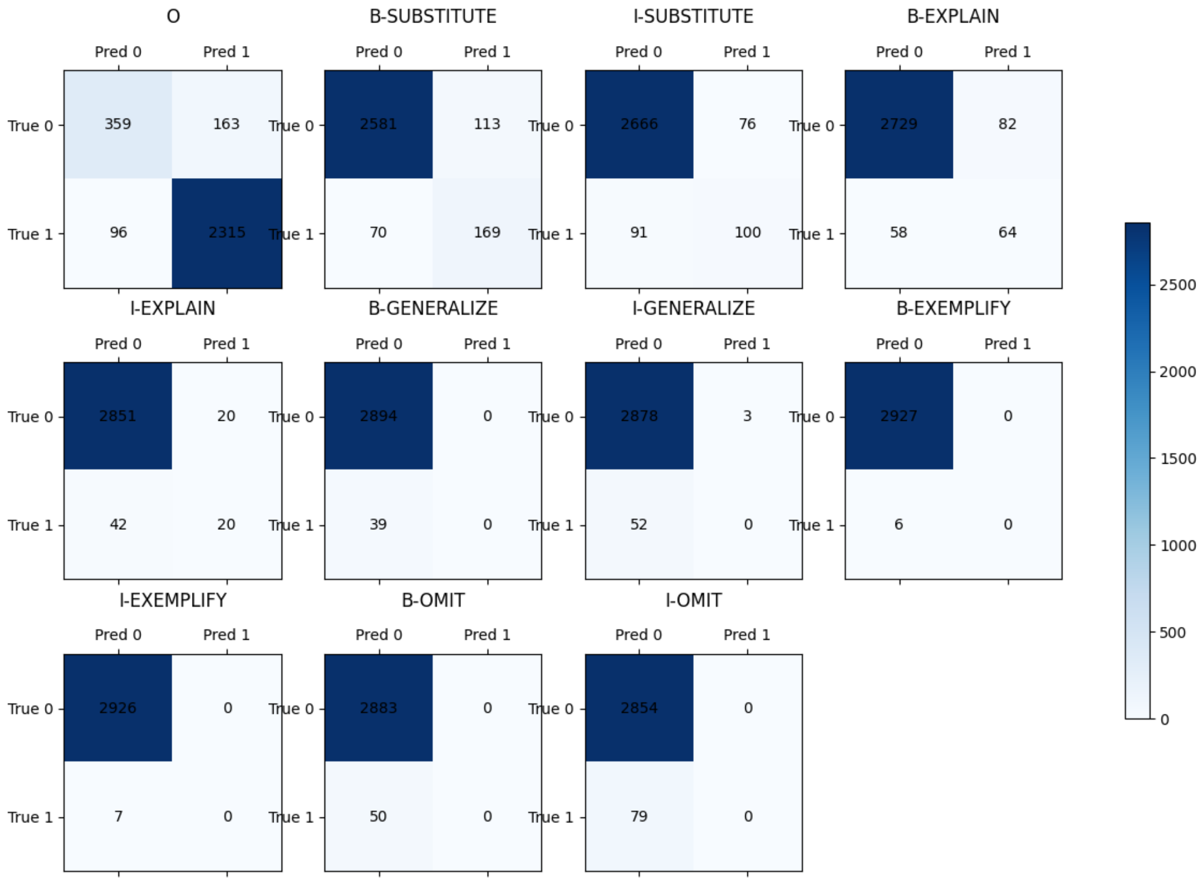

Figure 3: The confusion matrices for the results on the test set of Task-1B for each label.

	F1-micro	Precision-micro	Recall-micro
word level(w/ ['O'] label)	0.836	0.858	0.819
word level(w/o ['O'] label)	0.413	0.546	0.417

Table 2: System performance on the test set for task 1B.

limited training instances for these labels appear insufficient for effective model fine-tuning. Analysis of the training process reveals that the model rarely correctly classified these minority labels, suggesting room for improvement in handling class imbalance.

3 PLABA 2024 Task 2: Complete Abstract Adaptation

3.1 Introduction

Task 2 involves the end-to-end adaptation of biomedical abstracts for a general audience by utilizing plain language. Given a collection of abstracts as the source material, the system will generate a corresponding plain language adaptation as the output for each sentence within the source text.

3.2 Methodology

To generate simplified versions of abstract sentences, we employed the Llama-3.1-70B (dub, 2024) instruction tuned version [3] and GPT-4o (ope, 2024) with one-shot prompt only.

3.2.1 Corpus

In this shared task, we stick with the same PLABA corpus (Attal et al., 2022), which is introduced in section 2.2.1, but it is mostly used for evaluation only. In the corpus, there are 40 abstracts containing 7604 sentences in total with a reference or references.

[3] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

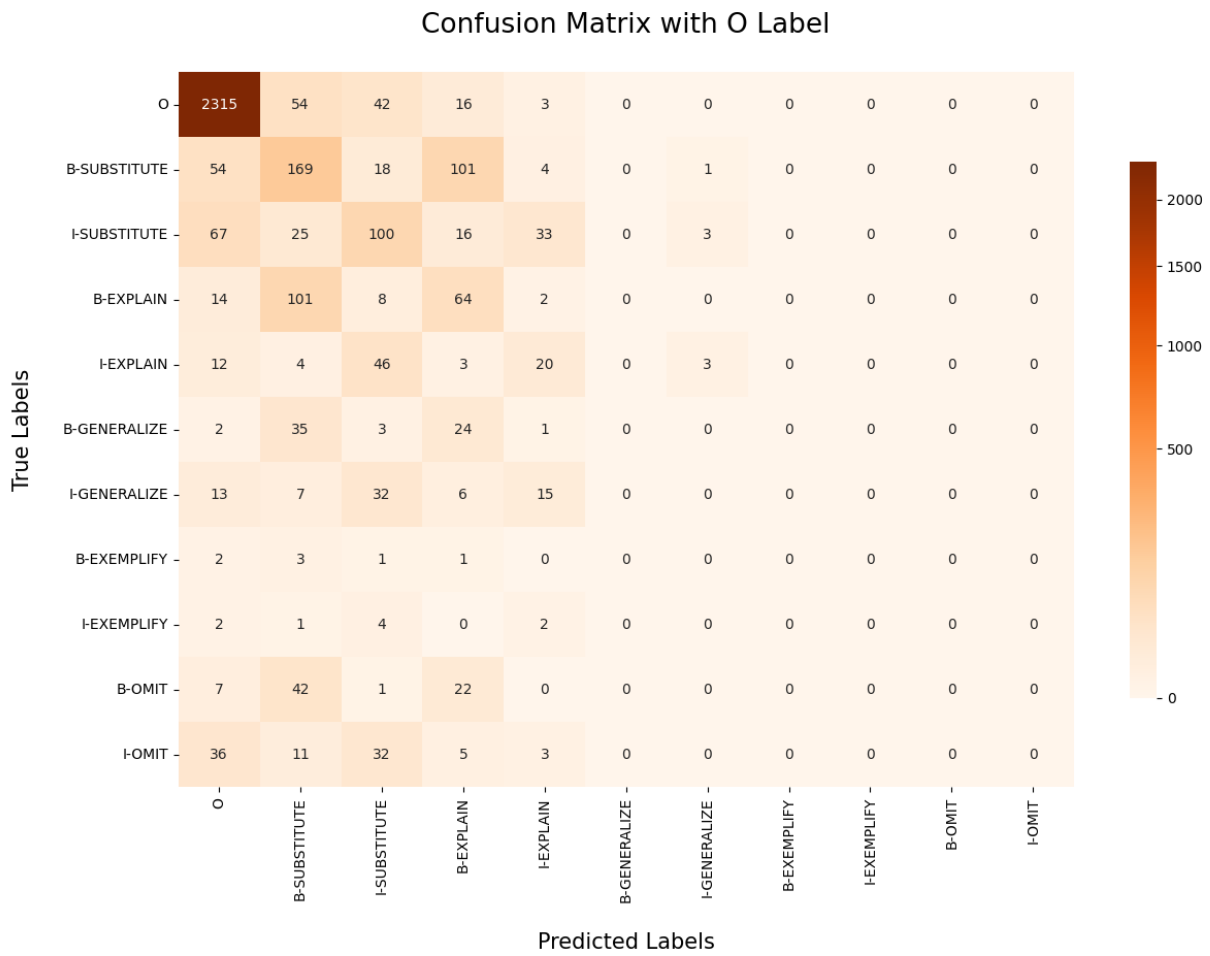

Figure 4: The confusion matrix between all labels on the test set for Task 1B.

3.2.2 Metrics

In this shared task, we mainly rely on automatic evaluation metrics, which can be classified into reference-based and reference-less metrics. In reference-based metrics, BLEU (Papineni et al., 2002), SARI (Xu et al., 2016), BERTScore (Zhang* et al., 2020) and LENS (Maddela et al., 2023) are chosen due to their popularity in text simplification areas. SALSA (Heineman et al., 2023), which is a new-developed metric, is used for reference-less evaluation.

BLEU The Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002) is a metric widely used in machine translation to evaluate the quality of the generated text. As text simplification can be regarded as monolingual text translation, it is also accepted and used in the early stages of text simplification studies. It measures the n-gram (from 1 to 4) text similarity between the generated text and the single or multiple references.

SARI The System Output Against Reference and Input (SARI) score (Xu et al., 2016) is a metric designed to evaluate text simplification tasks. Similar to BLEU, SARI also measures the n-gram similarity between the output and references. Additionally, the score is determined based on three operations: adding, deleting, and retaining words from the original sentence. Each operation is assessed by comparing it to one or more reference simplified texts to measure how closely the simplified text matches expert simplifications. As a result, it better aligns with human evaluations in text simplification. simplification.

BERTScore Unlike the two above-mentioned metrics, BERTSCore (Zhang* et al., 2020) measures the semantic similarity between the output and reference instead of n-gram similarity. By leveraging the deep contextualized embeddings from BERT (Devlin et al., 2019), it can measure the semantic similarity among the tokens. Then it calculates the sentence similarity by token matching and maximising the matrix between output and references.

LENS Learned Evaluation of Non-Canonical Sequences (LENS) (Maddela et al., 2023) is yet another metric for evaluating text generation tasks. The embedded LENS machine learning model in LENS makes it different from the previous metrics. It evaluates the generated text based on semantic and contextual similarity, which improves the adaptability and accuracy in tasks where the generated text may differ in surface form from reference text but still be semantically correct.

SALSA **S**uccess and F**A**ilure driven **L**inguistic **S**implification **A**nnotation (SALSA) (Heineman et al., 2023) is an edit-based human annotation framework. By training LENS (Maddela et al., 2023) on 19 thousand edit annotations gathered through SALSA, the authors developed LENS-SALSA, which we refer to simply as SALSA in this report. The main difference between SALSA and LENS is that SALSA can measure performance based on self-predicted references, which is beneficial for evaluating the test set.

3.3 Experiment

3.3.1 prompting process

To generate simplified sentences in the abstract, we started with a simple 1-shot prompt that has one example in the prompt. Then based on the output of test cases, we kept asking the model to self-adjust the prompt until it almost generated the expected outcome. We list the prompts we used in Tables 8 and 9 in the appendix.

3.3.2 post-processing

Sometimes models will output some pre-words such as "Here's a simplified version: " and "**Simplified**:", we checked the outputs and used the rule to delete these pre-words. Some outputs have detailed explanations for how and why did I simplify this sentence. For this kind of output, we used regular expressions to extract the simplified sentence as possible.

3.4 Results and Discussion

Automatic Evaluation Tables 3 and 4 show the performance difference between the two models across the metrics. In Table 3, all four metrics consistently favour GPT-4o over Llama and shows some credibility of SALSA as a reference-less metric. Similarly in Table 4, the two models show an aligned performance gap, compared to Table 3.

Model	SALSA	LENS	SARI	BERT(F_1)
Llama 3.1 instruct-70B	73.89	52.79	36.37	0.88
GPT-4o	74.28	64.75	37.72	0.92

Table 3: System performance on the evaluation set across SALSA, LENS, SARI and BERT for BERTScore.

Model	SALSA
Llama 3.1 instruct-70B	61.79 $\pm$ 0.30
GPT-4o	73.08 $\pm$ 0.24

Table 4: System performance on the test set under the SALSA (95% confidence interval).

4 Official Ranking from PLABA2024

4.1 Task-1 (A and B)

Table 5 shows the system ranking from PLABA2024 official organisers on Task 1A and 1B using F1 scores, 1C being not popular among the submissions. There are overall 9 systems evaluated by PLABA 2024 (Ondov et al., 2024). Out of the 9 systems, our team UM ranks the third for Task 1A, and the 2nd for Task 1B via F1 scores. There are a few intersting findings from the results.

- Firstly, it is worth to note that *for both tasks, our model scores are very close to the highest system*, e.g. for Task 1A, our F1 score is 0.4787 using Roberta-Base, after the first two systems 0.5036 from ReBERTa-GBC and 0.4885 from Gemini-1.5. For Task 1B, our F1 score 0.7765 is even closer to the top-1 score 0.7788 from MLPClassifier system. In general, our fine-tuned system is almost as good as the top-1 system at identifying the complicated term, but we are much better at classifying the next to-do step in the term processing (11 labels = 1 + 5x2) compared to the top-1 system for Task-1A (0.7765 vs 0.6838) winning almost 10 absolute point difference. It is also interesting to see that the Top-1 system for Task-1B (F1 0.7788) from MLPClassifier, produced the lowest F1 score for Task-1A, i.e. 0.0459, which means it actually cannot identify most of the complicated terms (only made into 5%).

- Secondly, our *averaged F1 score across Task 1A and 1B* is the **highest** among all 9 systems, 0.6276=(0.4787+0.7765)/2. It is also the only system whose Averaged F1 score is above 0.60.

- Thirdly, It is interesting to see that **much smaller models can outperform the extra-large LLMs** on Task 1A and 1B, e.g. our RoBERTa-base fine-tuned wins Mistral, GPT, Gemini-1.5 on these two tasks. This is related to the findings from (Li et al., 2024) for the PLABA 2023 shared task.

4.2 Task-2

For Task 2, even though GPT4o outscored LLaMa-3.1-70B in our model development phase, to understand better how open-science LLMs perform on biomedical text simplification tasks, we selected LLaMa-3.1-70B as our prioritised output over the GPT4o-instruct for official human evaluation in the PLABA-2024. As shown in Table 6, it produced interesting outcomes with the **highest** Completeness score (0.9481) vs the *lowest* Accuracy score (0.6088).

According to the human evaluation guidelines from PLABA-2024, Accuracy is “Outputs should contain the accurate information”, while Completeness is “Outputs should seek to minimize information lost from the original text”. So, LLaMa-3.1-70B-instructed can do its best to avoid information loss among the 8 evaluated systems, but it did not produce the most accurate outputs. These are somehow contradicting findings but it might be because the model produced some misleading outputs in addition to keeping the source information. We need further qualitative analysis on this assumption.

5 Conclusion and Future Work

In this report, we present the two systems submitted by the BeeManc team for two different tasks in PLABA 2024. For Task 1, we submitted a classification system based on Roberta for Task 1a and 1b. The system’s performance at the word level was not outstanding, highlighting the limitations of simple classification models in multi-label, multi-class tasks. For Task 2, we submitted end-to-end generation results based on LLaMa 3.1 and GPT-4o for the generation task. On the Evaluation set, GPT-4o performed slightly better than LLaMa. In the unsupervised evaluation on the test set, GPT-4o also outperformed LLaMa 3.1. Our findings demonstrate the potential of open-source models in text simplification tasks within the medical domain.

Regarding LLaMa-3.1-70B-instructed model producing the highest Completeness score but the lowest Accuracy according to the official human evaluation, we plan to carry out further qualitative analysis on its output to see if the system output contains misleading or hallucinated information in addition to the secured source knowledge. We can also use some specific human-in-the-loop metrics such as HOPE (Gladkoff and Han, 2022) to categorise the error types.

6 Acknowledgment

We thank Nicolo and Sam for their valuable discussion and feedback during this project with their expertise from the experience of PLABA2023. The authors wrote the initial draft, used Generative AI to revise/polish it, then the authors reviewed the generated content. The authors used ChatGPT to refine the paper writing.

References

2024. Gpt-4 technical report.

2024. The llama 3 herd of models.

Kush Attal, Brian Ondov, and Dina Demner-Fushman. 2022. A dataset for plain language adaptation of biomedical abstracts.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Serge Gladkoff and Lifeng Han. 2022. HOPE: A task-oriented and human-centric evaluation framework using professional post-editing towards more effective MT evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 13–21, Marseille, France. European Language Resources Association.

Team	**Run**	**Rank**	**Task 1A (F1)**	**Task 1B (F1)**	**Average Score**
BU	MLPClassifier-identify-classify-replace-v1	1	0.0459	**0.7788**	0.4124
CLAC	mistral	1	0.4410	0.6663	0.5537
CLAC	gpt	2	0.3767	0.3795	0.3781
IITH	First	1	0.1956	0.7014	0.4485
UM	Roberta-base	1	*0.4787*	*0.7765*	**0.6276**
ntu_nlp	gemini-1.5-pro_demon5_replace-demon5	1	*0.4885*	0.6335	0.5610
ntu_nlp	gemini-1.5-flash_demon5_replace-demon5	2	0.4431	0.6544	0.5488
ntu_nlp	gpt-4o-mini_demon5_replace-demon5	3	0.4518	0.6197	0.5358
Yseop	roberta-gbc	1	**0.5036**	0.6838	0.5937

Table 5: Task 1A and Task 1B (F1) Results: Roberta-base highest Averaged-Score(Ondov et al., 2024)

Run	**Accuracy**	**Completeness**	**Simplicity**	**Brevity**	**Final avg.**
LLaMA-8B-4bit-MedicalAbstract-seq-to-seq-v1	0.8545	0.9180	0.6533	0.3978	0.7059
LLaMa 3.1 70B instruction 2nd run	0.6088	**0.9481**	0.5561	0.5004	0.6533
TREC2024_SIB_run3	0.7506	0.7716	0.7997	0.6484	0.7426
UAms-ConBART-Cochrane	0.9534	0.9398	0.6851	0.6171	0.7988
gpt35_dspy	0.9167	0.9386	0.6974	0.5478	0.7751
mistral-FINAL	0.6694	0.7552	0.5149	0.2875	0.5567
plaba_um_fhs_sub1	0.9067	0.9138	0.7814	0.6799	0.8205
task2_moa_tier1_post	0.8982	0.9444	0.7246	0.5284	0.7739

Table 6: PLABA 2024 Task 2 results: LLaMa-3.1 with the highest Completeness (Ondov et al., 2024)

David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. 2023. Dancing between success and failure: Edit-level simplification evaluation using SALSA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3495, Singapore. Association for Computational Linguistics.

Zihao Li, Samuel Belkadi, Nicolo Micheletti, Lifeng Han, Matthew Shardlow, and Goran Nenadic. 2024. Investigating large language models and control mechanisms to improve text readability of biomedical abstracts. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, pages 265–274.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16383–16408, Toronto, Canada. Association for Computational Linguistics.

Brian Ondov, Bill Xia, Ishita Unde, Hoa T. Dang, and Dina Demner-Fushman. 2024. Overview of the trec 2024 plaba track. In *The Thirty-Third Text REtrieval Conference Proceedings (TREC 2024), Forthcoming*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pablo Romero, Lifeng Han, and Goran Nenadic. 2024. Insightbuddy-ai: Medication extraction and entity linking using large language models and ensemble learning.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. volume 4, pages 401–415.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Argument	Value
evaluation_strategy	`epoch`
num_train_epochs	10
weight_decay	0.01
save_strategy	`epoch`
learning_rate	0.00005
per_device_train_batch_size	32
per_device_eval_batch_size	32
dataloader_num_workers	4
dataloader_pin_memory	`True`
optim	`adamw_torch`
fp16	`True`
warmup_ratio	0.1

Table 7: Fine-tuning arguments for model training for RoBERTa-base

Section	Content
Objective	You are tasked with simplifying a complex medical text to make it easily understandable by non-medical professionals, such as patients or caregivers.
Guidelines	• **Maintain Accuracy:** Ensure that the simplified text accurately reflects the original meaning. Do not omit critical details or introduce incorrect information. • **Use Plain Language:** Replace medical jargon and complex terms with simple, everyday language. If a technical term is necessary, provide a brief explanation or analogy. • **Shorter Sentences:** Break down long, complicated sentences into shorter, more manageable ones. Multiple sentences are allowed but must fit within a single line per numbered point. • **Use Active Voice:** Where possible, use the active voice to make the text more direct and easier to read. • **Provide Examples:** Use examples or analogies to help explain complex concepts, making them relatable to everyday experiences. • **Simplify Structure:** Organize the information in a logical, easy-to-follow structure. Use bullet points or lists to highlight key points when applicable.
Output Guidelines	1. Align the output strictly with the numbered points from the original text. 2. Maintain the numbering format exactly as it appears in the input. 3. Each numbered point must be a single line but can include multiple sentences if necessary for clarity. 4. Ensure that each simplified point corresponds directly to its numbered counterpart in the original text.

Continued on next page

Section	Content
Original Text Example	1. The dystonias are a group of disorders characterized by excessive involuntary muscle contractions leading to abnormal postures and/or repetitive movements. 2. A careful assessment of the clinical manifestations is helpful for identifying syndromic patterns that focus diagnostic testing on potential causes. 3. If a cause is identified, specific etiology-based treatments may be available. 4. In most cases, a specific cause cannot be identified, and treatments are based on symptoms. 5. Treatment options include counseling, education, oral medications, botulinum toxin injections, and several surgical procedures. 6. A substantial reduction in symptoms and improved quality of life is achieved in most patients by combining these options.
Simplified Text Example	1. Dystonias are disorders with a lot of uncontrollable muscle contractions. This causes awkward postures and/or repetitive movements. 2. Checking the symptoms helps spot patterns and guides testing to find possible causes. 3. If a cause is found, treatments specific to that cause may be available. 4. When no specific cause is found, treatments focus on managing symptoms. 5. Treatment options include counseling, education, oral medications, botox (a muscle relaxant), and surgeries. 6. Combining these treatments usually leads to a noticeable decrease in symptoms and a better quality of life for most patients.

Table 8: Prompt used for GPT4o

Section	Content
Objective	Simplify the provided text by: 1. Rephrasing complex sentences for clarity. 2. Replacing or defining rarely-used terms.
Guidelines	- For sentences that seem complex, rephrase them in simpler terms. - If you encounter unfamiliar or rare words, either replace them with a commonly known synonym or provide a concise definition. Note: In the training samples, complex sentences are flagged with ‘ <rephrase >‘ and rare terms with ‘ <rare >‘. However, these tokens won’t appear in testing samples. You’ll need to recognize and address such complexities independently. Examples:

Continued on next page

Section	**Content**
Original Text Example	1.Original: 1. Furthermore, the circumference of thighs was measured to assess the <rare >postoperative swelling <rare >. 2. A total of 444 hypertensive patients, aged between 27 to 65 years, without any recent hypertensive treatment, were included. 3. <rephrase >The tongue often obstructs the upper respiratory tract, especially in comatose patients or those with cardiopulmonary arrest. <rephrase >
Simplified Text Example	1. Additionally, we measured thigh sizes to check for swelling after surgery. 2. 444 patients, aged 27-65 with high blood pressure and no recent treatment, were studied. 3. The tongue can block breathing, mostly seen in unconscious people or those who've had a sudden heart stoppage.

Table 9: Prompt used for LLaMa-3.1-70B instruct