Acta Cybernetica — *online–first paper version* — pages 1–16.

Optimizing Abstractive Arabic Summarization via RLHF and DPO with Llama 2

Mram Kahla[ab] and Zijian Győző Yang[cd]

Abstract

Given the advantages observed with Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO) in English, it is promising to explore their effectiveness for abstractive summarization in languages with complex morphological and syntactic features, such as Arabic. In this study, we fine-tune the Llama 2 model, which demonstrates a significant capability to enhance summarization results. We highlight how Llama 2, combined with advanced techniques like RLHF and DPO, markedly improves the quality of Abstractive Arabic summarization, showcasing the model's superior performance in this challenging task. Furthermore, the AraSum corpus plays a critical role in achieving outstanding results, highlighting its effectiveness in improving the performance of summarization models. While this work focuses on Arabic, the techniques and insights presented are language-agnostic, offering broader applications for abstractive summarization in other languages. Additionally, we introduce the AraRLHF and AraDPO datasets, which are publicly available to support reproducibility and advance research in Arabic NLP.

Keywords: abstractive summarization, Arabic, reinforcement learning, Direct Preference Optimization, RLHF, DPO, Llama 2

1 Introduction

In Natural Language Processing (NLP), automatic text summarization stands as a pivotal task, catering to the ever-increasing volume of information available in today's digital age.

Unlike extractive summarization [38] which selects and rephrases existing segments from the original text, abstractive summarization [28] involves generating novel sentences that capture the essence of the source material. This process

[a]Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, Hungary
[b]E-mail: kahla.mram@itk.ppke.hu, ORCID: 0000-0001-9885-8184
[c]HUN-REN Hungarian Research Centre for Linguistics, Budapest, Hungary
[d]E-mail: yang.zijian.gyozo@nytud.hun-ren.hu, ORCID: 0000-0001-9955-860X

DOI: 10.14232/actacyb.316434

demands a deep understanding of semantics, contextual nuances, and linguistic structures to produce coherent and concise summaries. Specifically in the context of the Arabic language, abstractive summarization holds significant promise and challenges due to the language's intricate syntax, rich morphology, and diverse linguistic features.

In recent years, reinforcement learning (RL) has emerged as a promising paradigm for enhancing sequence generation tasks in NLP [22], such as abstractive summarization and question-answering as a promising paradigm. RL enables models to align outputs with human preferences [40] and leverage human feedback to improve factual accuracy and user alignment [23]. With its ability to learn optimal decision-making policies through interaction with an environment, RL offers an effective approach to refining abstractive summarization models, particularly when the goal is to align generated summaries with human preferences.

The outcomes obtained in the English language summarization through Reinforcement Learning from Human Feedback (RLHF) demonstrate significant improvements in the quality of the generated text [29] offering a clear advantage over larger supervised models that rely solely on traditional training methods.

While RLHF has proven effective in adjusting model outputs to better reflect human preferences, it is not without its limitations. A major limitation of RLHF is that its process is considerably more complex than traditional supervised learning. To address this complexity, methods like Direct Preference Optimization (DPO) [27] have been introduced as simpler training paradigms. DPO enables language models to be trained from human preferences without the added complexity of reinforcement learning while performing as well as or even better than existing RLHF algorithms.

The objective of this research is to explore the application of Reinforcement Learning from Human Feedback and Direct Preference Optimization to the task of abstractive Arabic text summarization.

Our main contribution lies in applying Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO) to the task of abstractive text summarization for the Arabic language. We demonstrate how LLaMA 2, when combined with these advanced techniques and the AraSum corpus, significantly enhances the quality of Arabic text summarization. To foster reproducibility and encourage further research in Arabic NLP, we release the AraRLHF and AraDPO datasets, which consist of human preference data specifically tailored for RLHF and DPO models. The datasets are available on our GitHub[1].

The rest of the paper is structured as follows: Section 2 reviews related work. Section 3 outlines the methodology. Section 4 discusses the corpora used, and Section 5 details the models used. Section 6 presents our experiments and results, and finally, Section 7 concludes the paper.

[1] https://github.com/ppke-nlpg/AraSum

2 Related work

Reinforcement learning from human feedback (RLHF), originally developed for training simple robots in simulated environments and Atari games [8, 14].

In terms of reinforcement learning with human feedback to train text summarization models, Böhm et al. [4] learn a reward function from 2,500 human judgments of CNN/DM [24] summaries that are used in a reinforcement learning setting.

A similar method of recursive task decomposition was used for summarizing books [34]. They combine learning from human feedback with recursive task decomposition by using models trained on smaller parts of the task to assist humans in giving feedback on the broader task.

Ziegler et al. [40] fine-tune pre-trained language models with reinforcement learning by exploiting a reward model trained from human preferences. Then the model is used to generate summaries over Reddit TL;DR, and CNN/DM datasets. The limitation of their framework is that their labelers prefer extractive summaries and there are low agreement rates between labelers and researchers.

Stiennon et al. [29] followed their previous work on learning from human feedback and propose to gather a dataset composed of human preferences between pairs of summaries as the first step. Then the prediction of the human-preferred summary is generated by a reward model (RM) trained via supervised learning. Lastly, the score produced by the RM is maximized as much as possible by a policy trained via reinforcement learning. This approach significantly outperforms both human reference summaries and much larger models fine-tuned with supervised learning alone.

Although RLHF has proven effective in aligning model outputs with human preferences, it has certain limitations, such as the high cost and complexity of training reward models, and the potential for misalignment between the reward model and human preferences [5]. To address the complexity of RLHF optimization, Rafailov et al. [27] introduced Direct Preference Optimization (DPO) as an alternative approach. Unlike RLHF, DPO eliminates the need for training a reward model and instead directly trains the language model based on human preferences using a simple binary cross-entropy objective.

Human feedback has been utilized to improve various AI systems across different tasks. For instance, in dialogue systems, Jaques et al. [15] employed crowd-sourced human labeling to judge whether dialogue generated by an offline RL agent was fluent and amicable. Similarly, in the translation task, Kreutzer et al. [19] collected both explicit and implicit human feedback to improve a machine translation model by integrating the feedback into a reinforcement learning framework. In review generation, Cho et al. [6] developed models of coherence from existing texts and used these models as RL rewards to enhance long-form generation. For question-answering, Nakano et al. [23] fine-tuned GPT-3 to answer long-form questions within a web-browsing environment. This setup enabled the model to navigate the web and incorporate human feedback to optimize answer quality through imitation learning. Additionally, human feedback has been applied to other tasks,

such as evidence extraction [26], story generation [39], and semantic parsing [20].

The successful integration of RLHF into language technology was notably advanced by the development of ChatGPT [25]. This research achieved significant improvements in the model's ability to generate responses that align more closely with human-like communication. The approach began with a supervised fine-tuning phase, where the large language model was trained on prompts containing specific instructions. This was followed by an additional fine-tuning phase using reinforcement learning, further enhancing the model's response quality and alignment with human preferences.

Regarding abstractive summarization in the Arabic language, one study [3] introduced a four-stage abstractive summarization framework where the core of the system is an extractive summarizer. Training a model specifically for headline generation was presented in [1]. Another research [11] utilized the PreSumm approach along with a multilingual BERT model for fine-tuning both extractive and abstractive models. AraBART [18] is a pre-trained encoder-decoder model designed for abstractive summarization tasks tailored to the Arabic language. Furthermore, analysis by Chouikhi and Alsuhaibani [7] conducted a comparison analysis of various Arabic language models' performance in the task of text summarization.

There are two additional experiments conducted as part of the abstractive Arabic summarization task. In the first experiment, Kahla et al. [17] created the first monolingual, human-written corpus for abstractive Arabic text summarization and used it to fine-tune several language models: m-BERT, AraBERT, and m-BART-50. To enhance the performance of the baseline systems, a cross-lingual knowledge transfer method was applied. In the second experiment [16], they extended the Arabic summarization corpus, AraSum[2], and made it publicly available. This expanded corpus contains approximately 50,000 Arabic articles with their corresponding leads. The experiment involved pre-training monolingual and trilingual BART models for Arabic, as well as fine-tuning these models and the mT5 model for abstractive summarization using the AraSum corpus. Results showed that the models trained on AraSum performed well, surpassing the state-of-the-art XL-Sum [12] model at the time of publication.

In terms of Reinforcement Learning from Human Feedback and Direct Preference Optimization for the Arabic language, there is a noticeable scarcity of existing research. Leveraging RLHF and DPO presents a powerful technique that deserves application within such complex linguistic contexts.

3 Methodology

This research explores the application of RLHF and DPO to the task of abstractive Arabic text summarization.

[2] https://github.com/ppke-nlpg/AraSum

3.1 Reinforcement Learning experiments

The RLHF approach we adopt is based on OpenAI's methodology [29], consisting of three main steps:

- **Step 1: Collect demonstration data, train a supervised policy, and send comparisons to humans.**

  Humans are provided with reference texts and summaries generated by fine-tuning a language model. They are then asked to choose the best summary from the given samples.

- **Step 2: Collect comparison data, and train a reward model (RM).**

  A reward model is trained using the human feedback collected in the first step. Based on the annotations provided by the human evaluators, this model predicts the likelihood (log odds) that a given summary is preferred.

- **Step 3: Optimize a policy against the reward model using Proximal Policy Optimization (PPO).**

  The output of the reward model serves as a reward measure. The supervised policy will be fine-tuned to maximize this reward using reinforcement learning, with the Proximal Policy Optimization (PPO) algorithm guiding the optimization process.

3.2 Direct Preference Optimization experiments

For the DPO approach, we adopt the method proposed by Rafailov et al. [27], which simplifies the RLHF process by eliminating the need to fit a reward model. Instead, DPO directly trains language models based on human preferences. The DPO approach consists of the following steps:

- **Step 1: Collect preference data from human evaluators.**

  Human evaluators are provided with multiple summaries for a given input and asked to select the one they prefer.

- **Step 2: Apply Direct Preference Optimization.**

  DPO bypasses the need for a reward model and directly utilizes the human preference data to train the language model. The model is optimized by applying a binary cross-entropy objective, where it learns to assign higher probabilities to the summaries preferred by the human evaluators.

- **Step 3: Fine-tune the language model based on preferences.**

  The language model is fine-tuned to generate summaries that better align with human preferences, achieving this without the need for reinforcement learning algorithms.

4 Corpora used

For our experiments, two datasets are required: The first dataset is used in RLHF to train the reward model to assess summary quality, while in DPO, it directly guides the optimization of the language model based on human preferences. The second dataset is used in the final step of both methodologies, which involves fine-tuning the models based on the collected preferences.

4.1 Human Preference Dataset

The first dataset, named AraRLHF and AraDPO, is utilized in the initial step of both RLHF and DPO, focusing on collecting preference data from human evaluators. This dataset is then employed in Step 2 of each methodology. In RLHF, the AraRLHF dataset is used to train the reward model (RM), which predicts the quality of generated summaries based on the collected human preferences. In DPO, the AraDPO dataset is used directly to train the language model based on these preferences, without the need for a reward model.

To create this dataset, we utilized manual evaluation results from our previous research [17], where we fine-tuned transformer models for abstractive Arabic text summarization using the first version of AraSum. This corpus includes 21,508 articles and their corresponding leads. The transformer models evaluated were as follows:

- m-BERT model [10]: fine-tuned for Arabic.
- AraBERT model [2]: fine-tuned for Arabic.
- m-BART-50 model [30]: fine-tuned for Arabic.
- m-BERT+HUN model [36]: originally fine-tuned for Hungarian and then fine-tuned for Arabic.
- m-BERT+ENG model: first fine-tuned for English and then fine-tuned for Arabic.
- m-BART-50+RUS model: first fine-tuned for Russian then fine-tuned for Arabic.

The evaluation involved three human evaluators who evaluated the outputs of these six models, indicating their preferred summaries for a given input by assigning scores to each summary from 100 random samples, see figure 1.

The human evaluation data underwent preprocessing and was restructured to be suitable for training the reward model in RLHF and for direct use in DPO. The AraRLHF dataset, used to train the reward model, consists of 1,746 samples. These were derived from the evaluated articles (97), each associated with outputs from 6 different models, evaluated by 3 human annotators. This setup resulted in 6 evaluation tasks per article $\times$ 97 articles = 582 tasks, and with 3 annotators per

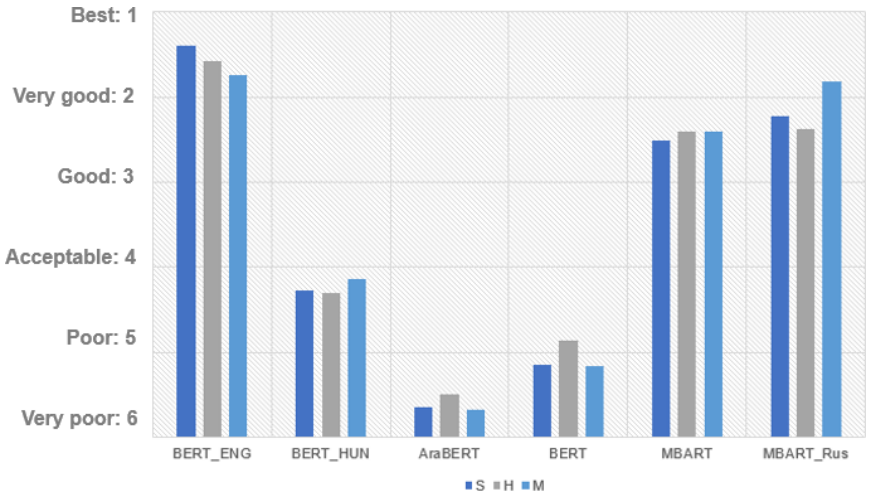


Figure 1: Human evaluation results from our previous study [17], where H, S, and M represent the human evaluators.

task, a total of 1,746 ranked preference records (582 × 3). Each record includes an article, its lead, and a label indicating the ranking of model outputs.

To construct the AraDPO dataset, we converted each ranked preference in AraRLHF into all possible pairwise comparisons. With 6 model outputs and 3 annotators, this yields 153 comparisons per article. Across all evaluated articles (97), this results in 14,841 comparisons, and since each comparison generates two records (preferred 1, and dispreferred 0), the final AraDPO dataset contains 29,682 binary preference records. Then we utilized a unique algorithm to avoid duplications. After deduplication, 2,309 records remained for training and testing. AraDPO contains all binary preference pairs derived from AraRLHF. Both datasets were randomly shuffled and split into 80% for training and 20% for testing.

The AraRLHF and AraDPO datasets are publicly available in JSON format to support reproducibility and encourage further research in Arabic NLP. The datasets can be accessed at the following link: https://github.com/ppke-nlpg/AraSum.

4.2 Dataset for Fine-tuning Llama 2

For fine-tuning Llama 2, we used the extended version of the AraSum corpus [16], which contains 49,604 articles along with their corresponding leads. In addition, we used the Arabic portion of the multilingual XL-Sum corpus [12], which consists of 46,897 articles and their corresponding leads. Both datasets are designed for abstractive text summarization.

For instruction fine-tuning, we used the prompt template recommended by the Stanford Alpaca research [31]:

> Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
> ### Instruction:
> Summarize the article written in Arabic below.
> ### Input:
> [article text]
> ### Response:
> [article summary]

Because the Llama 2 model is English-centric, we used an English template.

5 Experiments and Results

In our first experiment, we fine-tuned state-of-the-art Arabic summarization models with RLHF, specifically the mT5++ models from our previous research [16].

In the next experiment, we performed supervised fine-tuning (SFT) on the Llama 2 model for Arabic summarization. Following that, we applied RLHF and DPO fine-tuning to the SFT model.

Llama 2 [32] is an advanced large language model developed by Meta, marking the second iteration of the LLaMA series [33]. It represents a significant advancement in natural language processing. We used LLaMA 2 in this work as it was the most recent publicly available version at the time of experimentation. Llama 2 is available in various sizes: a 7-billion-parameter model, a 13-billion-parameter model, and a 70-billion-parameter model. For our experiments, we fine-tuned the smallest model with 7 billion parameters. For supervised fine-tuning, we used the Stanford Alpaca implementation [31]. The training hyperparameters are as follows: learning rate = 2e-5; global batch size = 256; epoch = 3; warmup ratio = 0.03; sequence length = 1800; bf16; deepspeed. For this task, we utilized eight NVIDIA A100 GPUs, each with 80GB of memory.

For RLHF experiments, the Transformer Reinforcement Learning X implementation from CarperAI [13] has been applied. The training hyperparameters are as follows: learning rate = 1e-5; global batch size = 4; epoch = 3; sequence length = 1800; number layers unfrozen = 2.

For the reward model, we fine-tuned an XLM-RoBERTa-large [9] model for the Arabic summarization quality prediction model. For this task, we use the Hugging Face implementaion[3]. The training hyperparameters are as follows: learning rate = 2e-5; global batch size = 32; epoch = 10; sequence length = 1024. We also conducted experiments with the mT5 base and large models [35], but they only achieved a Pearson correlation of 10–20.

In Table 1, we can see the results of the reward model experiments. The evaluation metrics are the same as those used in the research by [37]: Pearson Correlation, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). We achieved

[3] https://github.com/huggingface/transformers/tree/main/examples/pytorch

the highest Pearson correlation of **88** with 5 epochs. We used this checkpoint in subsequent experiments.

Table 1: Reward model experiments

	Pearson correlation ↑	MAE ↓	RMSE ↓
XLM-RoBERTa-base	81.25	0.83	1.04
XLM-RoBERTa-large	**88.00**	**0.69**	**0.86**
mT5-base	10.73	1.53	1.75
mT5-large	21.49	1.77	1.97

For the DPO experiments, we utilized the Hugging Face implementation[4], which is based on the original DPO research [27]. The training hyperparameters are as follows: learning rate = 5e-4; global batch size = 16; epoch = 3; sequence length = 1800.

In both the RLHF and DPO experiments, we tested different hyperparameters, with the best ones described above. For these tasks, we utilized a single NVIDIA A100 GPU with 80GB of memory.

The models that were experimented with and evaluated are as follows:

- **mT5++:** The state-of-the-art mT5-small model from the study of [16], fine-tuned using the AraSum corpus and using the XL-Sum Arabic corpus.
- **'mT5++' + RLHF:** The fine-tuned mT5++ model is further fine-tuned with the RLHF approach, where a reward model is trained from human feedback, followed by Proximal Policy Optimization (PPO) for policy refinement on the AraSum corpus, and the XL-Sum corpus.
- **'mT5++' + DPO:** The fine-tuned mT5++ model is further fine-tuned with the Direct Preference Optimization (DPO) approach using the human-evaluated dataset.
- **Llama 2:** The Llama 2 model with 7 billion parameters, supervised fine-tuned (SFT) using the AraSum corpus, and the XL-Sum Arabic corpus.
- **Llama 2 + RLHF:** The SFT Llama 2 model fine-tuned with the RLHF approach using the development set of AraSum and XL-Sum Arabic corpus, and the fine-tuned XLM-RoBERTa-large reward model.
- **Llama 2 + DPO:** The SFT Llama 2 model is fine-tuned with the Direct Preference Optimization approach using the human-evaluated dataset.

We evaluated the system output using the ROUGE-N and ROUGE-L metrics. ROUGE-1 and ROUGE-2 assess the overlap of word unigrams and bigrams, respectively, while ROUGE-L measures the overlap of the longest common subsequence

[4] https://huggingface.co/docs/trl/dpo_trainer

between two texts. ROUGE-L sum extends this by applying the ROUGE-L calculation at the sentence level and then aggregating the results for the final score.

It should be noted that the specific ROUGE scores presented here were calculated using the latest version of the ROUGE [21] library that was implemented by Hugging Face[5]), with the following setting: use_stemmer=True. Using the latest version, we were unable to reproduce the original values published in [16] and [12]. We also tried using the implementation of XL-Sum[6] and the original implementation by Google[7], but neither worked. The main objective is to demonstrate the enhanced performance resulting from our experiments; therefore, we used the values from the latest version of the Hugging Face Evaluate library. For better readability, we used the ROUGE * 100 values, similar to the approach described in [12] and [16]. In future work, we plan to enhance the evaluation step by incorporating Arabic-specific stemming to improve the reliability and comparability of ROUGE scores. For better transparency, the old and new ROUGE values for the mT5++ models are presented as follows (in the order: ROUGE-1/ROUGE-2/ROUGE-L):

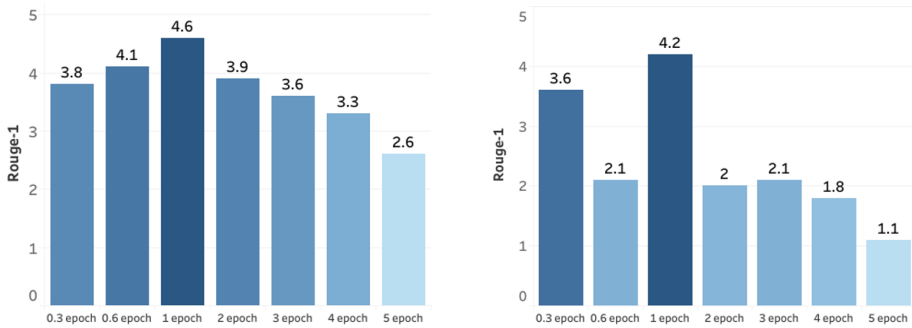
- old values of mT5++ Arasum Test Set: 33.172/13.914/24.782
- new values of mt5++ Arasum Test Set: 4.560/0.344/4.509
- old values of mT5++ XL-Sum Test Set: 29.128/11.049/24.070
- new values of mt5++ XL-Sum Test Set: 1.489/0.043/1.483

As with other fine-tuning experiments, we needed to determine the optimal number of epochs.

Figure 2 shows the performance of the fine-tuned models using RLHF and DPO across different training durations, ranging from 0.3 to 5 full epochs. The highest ROUGE scores were achieved at the 1 epoch mark for both methods (4.6 for RLHF and 4.2 for DPO). The labels "own dev RL" and "own DPO" refer to evaluations conducted on our dataset for the RLHF and DPO models.

Table 2 presents the experimental results. The ROUGE scores reveal several significant insights across the models and fine-tuning approaches. Llama 2, with its 7 billion parameters, significantly outperforms the mT5++ model across all metrics, demonstrating Llama 2's superior capabilities. Both RLHF and DPO contribute to improved performance, with Llama 2 + RLHF achieving the highest scores on the AraSum dataset, indicating a substantial boost in performance. Among all models, the 'mT5++' + DPO model performs the worst on most evaluation metrics, suggesting that mT5 does not benefit much from the DPO approach. The 'mT5++' + RLHF model performs slightly better but still worse than other optimized models. This suggests that mT5 may not respond well to optimization using human preference data. Additionally, the ROUGE scores for XL-Sum are significantly lower compared to AraSum across all models, highlighting the strength and quality of the AraSum dataset in achieving better summarization performance.

[5] https://huggingface.co/docs/evaluate/index
[6] https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring
[7] https://github.com/google-research/google-research/tree/master/rouge



(a) Own dev RL (b) Own DPO

Figure 2: **ROUGE-1 Scores for RLHF and DPO Fine-Tuning Across Different Training Epochs.** The left subplot shows the ROUGE-1 scores of models fine-tuned using RLHF, while the right subplot shows those fine-tuned using DPO. Both methods achieved their highest ROUGE-1 scores after one full epoch of training.

Table 2: ROUGE scores on the AraSum and the XL-Sum Arabic test sets.

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L sum
AraSum Test Set				
mT5++	4.560	0.344	4.509	4.537
'mT5++' + RLHF	3.464	0.245	3.435	3.444
'mT5++' + DPO	2.813	0.248	2.819	0.543
Llama 2	**4.636**	**0.414**	**4.618**	**4.616**
Llama 2 + RLHF	**4.947**	**0.486**	**4.957**	**4.949**
Llama 2 + DPO	**4.719**	**0.470**	**4.659**	**4.664**
XL-Sum Arabic Test Set				
mT5++	1.489	0.043	1.483	1.481
'mT5++' + RLHF	0.633	0.014	0.626	0.635
'mT5++' + DPO	0.534	0.029	0.540	0.543
Llama 2	**2.241**	**0.102**	**2.225**	**2.223**
Llama 2 + RLHF	**2.344**	**0.104**	**2.339**	**2.325**
Llama 2 + DPO	**2.447**	**0.112**	**2.440**	**2.431**

6 Conclusion

In this paper, we applied Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO) to the task of abstractive text sum-

marization for the Arabic language. By fine-tuning the state-of-the-art LLaMA 2 model, we observed a remarkable enhancement in summarization quality, particularly when RLHF was used with the AraSum dataset. The performance improvements highlight the strength of LLaMA 2, especially when combined with RLHF on our dataset. Moreover, the AraSum corpus played a crucial role in achieving superior results, consistently surpassing models fine-tuned on the XL-Sum dataset. This study demonstrates that advanced techniques like RLHF and DPO, in combination with a robust dataset such as AraSum and a highly capable large language model such as LLaMA 2, can significantly elevate the quality of abstractive Arabic text summarization.

While our focus was on Arabic, the techniques and insights presented in this study are inherently language-agnostic. They have the potential to be applied to other languages, particularly those with complex morphological and syntactic features, making this work relevant for broader multilingual NLP tasks. In addition, we publicly release the AraRLHF and AraDPO datasets to promote reproducibility and further advancements in Arabic NLP.

In the meantime, the Llama 3 models have been released. We aim to continue our experiments with these new models and anticipate achieving further advancements in performance through their utilization.

References

[1] Al-Maleh, M. and Desouki, S. Arabic text summarization using deep learning approach. *Journal of Big Data*, 7:1–17, 2020. DOI: 10.1186/s40537-020-00386-7.

[2] Antoun, W., Baly, F., and Hajj, H. AraBERT: Transformer-based model for Arabic language understanding. In Al-Khalifa, H., Magdy, W., Darwish, K., Elsayed, T., and Mubarak, H., editors, *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, 2020. European Language Resource Association. URL: https://aclanthology.org/2020.osact-1.2.

[3] Azmi, A. M. and Altmami, N. I. An abstractive Arabic text summarizer with user controlled granularity. *Information Processing and Management*, 54(6):903–921, 2018. DOI: 10.1016/j.ipm.2018.06.002.

[4] Böhm, F., Gao, Y., Meyer, C. M., Shapira, O., Dagan, I., and Gurevych, I. Better rewards yield better summaries: Learning to summarise without references. *arXiv Preprint*, 2019. DOI: 10.48550/arXiv.1909.01214.

[5] Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv Preprint*, 2023. DOI: 10.48550/arXiv.2307.15217.

[6] Cho, W. S., Zhang, P., Zhang, Y., Li, X., Galley, M., Brockett, C., Wang, M., and Gao, J. Towards coherent and cohesive long-form text generation. *arXiv Preprint*, 2018. DOI: 10.48550/arXiv.1811.00511.

[7] Chouikhi, H. and Alsuhaibani, M. Deep transformer language models for Arabic text summarization: A comparison study. *Applied Sciences*, 12(23), 2022. DOI: 10.3390/app122311944.

[8] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4302–4310, 2017. URL: https://dl.acm.org/doi/10.5555/3294996.3295184.

[9] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.747.

[10] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 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, 2019. Association for Computational Linguistics. DOI: 10.18653/v1/N19-1423.

[11] Elmadani, K. N., Elgezouli, M., and Showk, A. BERT fine-tuning for Arabic text summarization. *ArXiv Preprint*, 2020. DOI: 10.48550/arXiv.2004.14135.

[12] Hasan, T., Bhattacharjee, A., Islam, M. S., Mubasshir, K., Li, Y.-F., Kang, Y.-B., Rahman, M. S., and Shahriyar, R. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.findings-acl.413.

[13] Havrilla, A., Zhuravinskyi, M., Phung, D., Tiwari, A., Tow, J., Biderman, S., Anthony, Q., and Castricato, L. trlX: A framework for large scale reinforcement learning from human feedback. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 8578–8595, Singapore, 2023. Association for Computational Linguistics. DOI: 10.18653/v1/2023.emnlp-main.530.

[14] Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in Atari. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8022–8034, 2018. DOI: 10.48550/arXiv.1811.06521.

[15] Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, À., Jones, N., Gu, S., and Picard, R. W. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *CoRR*, abs/1907.00456, 2019. DOI: 10.48550/arXiv.1907.00456.

[16] Kahla, M., Novák, A., and Yang, Z. G. Fine-tuning and multilingual pretraining for abstractive summarization task for the Arabic language. *Annales Mathematicae et Informaticae*, 2023. DOI: 10.33039/ami.2022.11.002.

[17] Kahla, M., Yang, Z. G., and Novák, A. Cross-lingual fine-tuning for abstractive Arabic text summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 655–663. INCOMA Ltd., 2021. URL: https://aclanthology.org/2021.ranlp-1.74/.

[18] Kamal Eddine, M., Tomeh, N., Habash, N., Le Roux, J., and Vazirgiannis, M. AraBART: a pretrained Arabic sequence-to-sequence model for abstractive summarization. In Bouamor, H., Al-Khalifa, H., Darwish, K., Rambow, O., Bougares, F., Abdelali, A., Tomeh, N., Khalifa, S., and Zaghouani, W., editors, *Proceedings of the The Seventh Arabic Natural Language Processing Workshop*, pages 31–42, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. DOI: 10.18653/v1/2022.wanlp-1.4.

[19] Kreutzer, J., Khadivi, S., Matusov, E., and Riezler, S. Can neural machine translation be improved with user feedback? *arXiv Preprint*, 2018. DOI: 10.48550/arXiv.1804.05958.

[20] Lawrence, C. and Riezler, S. Improving a neural semantic parser by counterfactual learning from human bandit feedback. *arXiv Preprint*, 2018. DOI: 10.48550/arXiv.1805.01252.

[21] Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/W04-1013.

[22] Liu, Y., Wang, H., Zhou, H., Li, M., Hou, Y., Zhou, S., Wang, F., Hoetzlein, R., and Zhang, R. A review of reinforcement learning for natural language processing, and applications in healthcare. *arXiv Preprint*, 2023. DOI: 10.48550/arXiv.2310.18354.

[23] Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv Preprint*, 2021. DOI: 10.48550/arXiv.2112.09332.

[24] Nallapati, R., Zhou, B., and Ma, M. Classify or select: Neural architectures for extractive document summarization. *arXiv Preprint*, 2016. DOI: 10.48550/arXiv.1611.04244.

[25] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 27730–27744, Red Hook, NY, USA, 2024. Curran Associates Inc. URL: https://dl.acm.org/doi/10.5555/3600270.3602281.

[26] Perez, E., Karamcheti, S., Fergus, R., Weston, J., Kiela, D., and Cho, K. Finding generalizable evidence by learning to convince Q&A models. *arXiv Preprint*, 2019. DOI: 10.48550/arXiv.1909.05863.

[27] Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing System*, pages 53728–53741, 2023. URL: https://dl.acm.org/doi/10.5555/3666122.3668460.

[28] See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, 2017. Association for Computational Linguistics. DOI: 10.18653/v1/P17-1099.

[29] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize from human feedback. *arXiv Preprint*, 2020. DOI: 10.48550/arXiv.2009.01325.

[30] Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv Preprint*, 2020. DOI: 10.48550/arXiv.2008.00401.

[31] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford Alpaca: An instruction-following LLaMA model, 2023. URL: https://github.com/tatsu-lab/stanford_alpaca.

[32] Touvron, H. et al. Laama 2: Open foundation and fine-tuned chat models. *arXiv Preprint*, 2023. DOI: 10.48550/arXiv.2307.09288.

[33] Touvron, H. et al. LLaMa: Open and efficient foundation language models. *arXiv Preprint*, 2023. DOI: 10.48550/arXiv.2302.13971.

[34] Wu, J., Ouyang, L., Ziegler, D. M., Stiennon, N., Lowe, R., Leike, J., and Christiano, P. F. Recursively summarizing books with human feedback. *CoRR*, abs/2109.10862, 2021. DOI: 10.48550/arXiv.2109.10862.

[35] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.naacl-main.41.

[36] Yang, Z. G., Agócs, Á., Kusper, G., and Váradi, T. Abstractive text summarization for Hungarian. *Annales Mathematicae et Informaticae*, 53:299–316, 2021. DOI: 10.33039/ami.2021.04.002.

[37] Yang, Z. G. and Laki, L. J. Enhancing machine translation with quality estimation and reinforcement learning. *Annales Mathematicae et Informaticae*, 58:180–190, 2023. DOI: 10.33039/ami.2023.08.008.

[38] Zhang, X., Lapata, M., Wei, F., and Zhou, M. Neural latent extractive document summarization. *arXiv Preprint*, 2018. DOI: 10.48550/arXiv.1808.07187.

[39] Zhou, W. and Xu, K. Learning to compare for better training and evaluation of open domain natural language generation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 34, pages 9717–9724, 2020. DOI: 10.1609/aaai.v34i05.6521.

[40] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv Preprint*, 2019. DOI: 10.48550/arXiv.1909.08593.