Softbank-Meisei at TREC 2024 Ad-hoc Video Search and Video to Text Tasks

Kazuya Ueki[1], Yuma Suzuki[2], Hiroki Takushima[2], Haruki Sato[3], Takumi Takada[4], Aiswariya Manoj Kumar[2], Hayato Tanoue[2], Hiroki Nishihara[2], Yuki Shibata[2], and Takayuki Hori[2]

[1] Department of Information Science, Meisei University,
Room 27-1809, Hodokubo 2-1-1, Hino, Tokyo 191–8506, Japan

[2] AI Strategy Office, AI&Data Technology Planning Division,
Technology planning department, SoftBank Corporation
1-7-1 Kaigan, Minato-ku, Tokyo 105–7529, Japan

[3] Agoop Corporation,
3-35-8 Jingumae, Shibuya, Tokyo 150-0001, Japan

[4] SB Intuitions Corporation,
1-7-1 Kaigan, Minato-ku, Tokyo 105–7529, Japan kazuya.ueki@meisei-u.ac.jp

Abstract. The Softbank-Meisei team participated in the ad-hoc video search (AVS) and video-to-text (VTT) tasks at TREC 2024. In this year's AVS task, we submitted four fully automatic systems for both the main and progress tasks. Our systems utilized pre-trained vision and language models, including CLIP, BLIP, and BLIP-2, along with several other advanced models. We also expanded the original query texts using text generation and image generation techniques to enhance data diversity. The integration ratios of these models were optimized based on results from previous benchmark test datasets. In this year's VTT, as last year, we submitted four main task methods using multiple model captioning, reranking, and generative AI for summarization. For the subtasks, we submitted three methods using the output of each model. Last year's test data for the main task showed improvements of about 0.04 points in CIDEr-D and about 0.03 points in SPICE, based on the indices we had on hand.

1 AVS Task

1.1 System Overview

Our system utilized multiple state-of-the-art pre-trained multimodal models, including CLIP [1], BLIP [2], and BLIP-2 [3], as well as several other advanced models, to improve the accuracy of video retrieval. We extracted frame images from videos by sampling approximately every 10 frames and computed the similarity between each extracted frame and the query. The videos were then ranked based on their similarity scores in descending order. To further enhance data diversity, we expanded the original query texts using advanced text generation techniques and generated the corresponding images using image generation models. These expanded query representations allowed for a broader exploration of potential matches between the query and visual content. Furthermore, the integration ratios of the models were optimized using results from previous benchmark test datasets, ensuring that each model contributed optimally to the overall performance of the system.

1.2 Pre-traiend Visuion and Language Multimodal Models

First, we adopted models based on CLIP (Contrastive Language-Image Pretraining) [1], which enables effective alignment between visual and textual data for multimodal tasks. CLIP's framework has proven to be highly effective in tasks requiring the integration of vision and language, making it a cornerstone for our video retrieval system.

In particular, we leveraged a wide range of models provided by OpenCLIP, a major platform offering numerous cutting-edge models for multimodal applications. Among the available models are SigLIP [4], EVA-CLIP [5], and MetaCLIP [6], which are examples of models trained on large-scale datasets and are well-suited for handling complex interactions between images and text. These vision and language models, in general, have been trained on massive captioned datasets such as LAION-2B and LAION-5B [7], which contain billions of image-text pairs. This extensive training allows the models to capture a wide range of semantic relationships, enhancing their generalization across diverse queries and visual content.

In addition to the models provided by OpenCLIP, we also explored and experimented with other advanced vision and language models, such as BLIP [2], BLIP-2 [3], ALIGN [8], Long-CLIP [9], VeCLIP [10], ViTamin [11], LLaVA [12], and Phi3-Vision [13]. Through detailed evaluation, we selected the models that demonstrated superior accuracy and performance in line with our system requirements.

1.3 Query Expansion

We utilized GPT-3.5 to expand the original queries. Specifically, we provided prompts that instructed the model to generate 10 alternative sentences that conveyed the same meaning as the original query but used different expressions. This process was repeated 10 times, resulting in a large pool of generated sentences. Afterward, we removed any duplicates, ensuring that only unique sentences were retained. To address the issue of generated sentences that diverged too far from the original meaning, we employed a multimodal model to evaluate the semantic similarity between the original query and the generated sentences. Only sentences with a high similarity score were kept, while those with a lower similarity were automatically discarded to maintain consistency with the original query.

For image generation, we employed both the Stable Diffusion v2.1 and Stable Diffusion XL 1.0 models [14] provided by Stability AI to generate multiple images from the expanded queries. These models allowed for the creation of diverse visual representations based on text inputs. However, to ensure that the generated images were contextually aligned with the original query, we again applied a multimodal model to measure the similarity between the original query and each generated image. Images that showed a low degree of relevance or deviated from the original query's context were automatically filtered out, leaving only those that were closely related to the query in both content and theme.

These expansions offer several advantages. First, by generating diverse yet semantically similar queries, we increase the coverage of potential matches during video retrieval, enhancing the system's ability to find relevant results that may not have been captured by the original query alone. In addition, the use of multiple image representations improves the robustness of the system, allowing it to handle a wider variety of visual content.

1.4 Optimization of Integration Ratios

Table 1. Priorities and increments for integration ratio optimization.

Priority	Details
1	Best integration ratio for tv22 + tv23 (increment: 0.1)
2	Best integration ratio for tv22 + tv23 (increment: 0.05)
3	Best integration ratio for tv22 (increment: 0.1)
4	Best integration ratio for tv23 (increment: 0.1)

To achieve optimal integration of the models, we performed normalization and increment-based optimization. The similarity scores for each model were normalized using min-max scaling to a range of 0.0 to 1.0, ensuring consistency across different models and datasets as a uniform basis for integration.

The integration ratios were optimized for both combined and individual models from the tv22 and tv23 datasets. For the combined models, increments of 0.1 were used initially, with finer adjustments in increments of 0.05 where necessary. The optimization process used forward selection, where the models were integrated one by one. At each step, the integration ratio for the selected model was determined, starting with the model achieving the highest mean Average Precision (mAP), which was initially adopted with an integration ratio of 1.0. Subsequent models were integrated incrementally, with ratios adjusted between 0.1 and 2.0 (or 0.05 and 2.0 for finer adjustments), and the performance was reassessed. This process was repeated until no further performance improvements could be achieved.

The priorities and increments used during this optimization are summarized in Table 1.

1.5 Submissions and Results

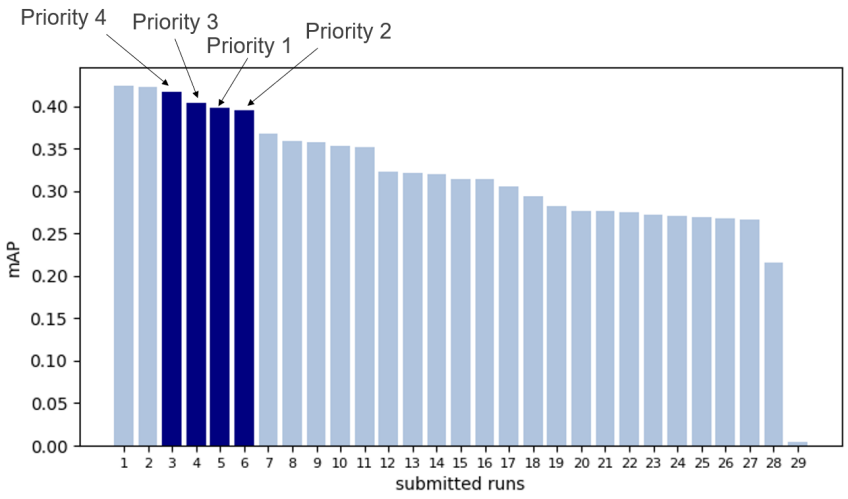

Fig. 1. Results of all fully automatic systems for all teams that submitted to the main task.

As shown in Fig. 1, our submitted systems ranked between 3rd and 6th among all submitted runs for the main task. This performance placed our team in the second overall among the participating teams.

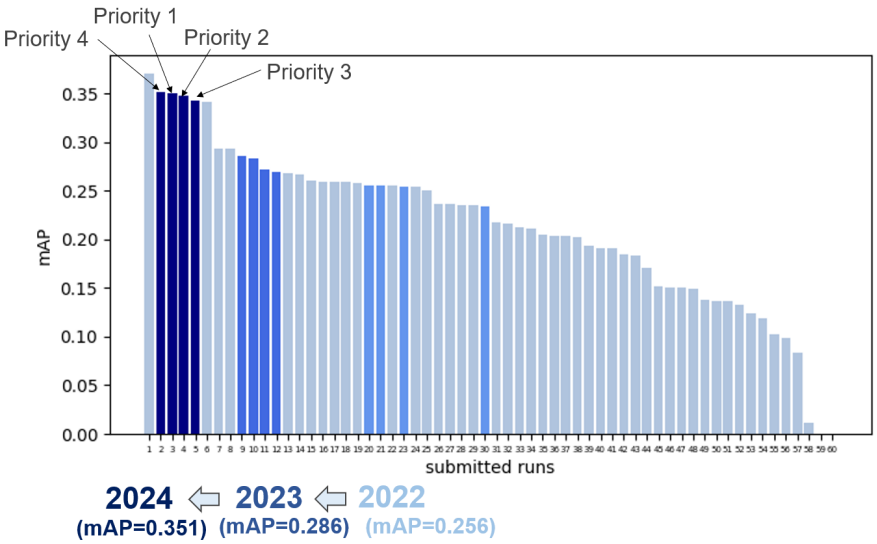

Fig. 2. Progress task results (Set-A). Our team ranked 2nd, with mAP improving from 0.256 (2022) to 0.286 (2023) and 0.351 (2024) due to advancements in visual-semantic embedding models and query expansion.

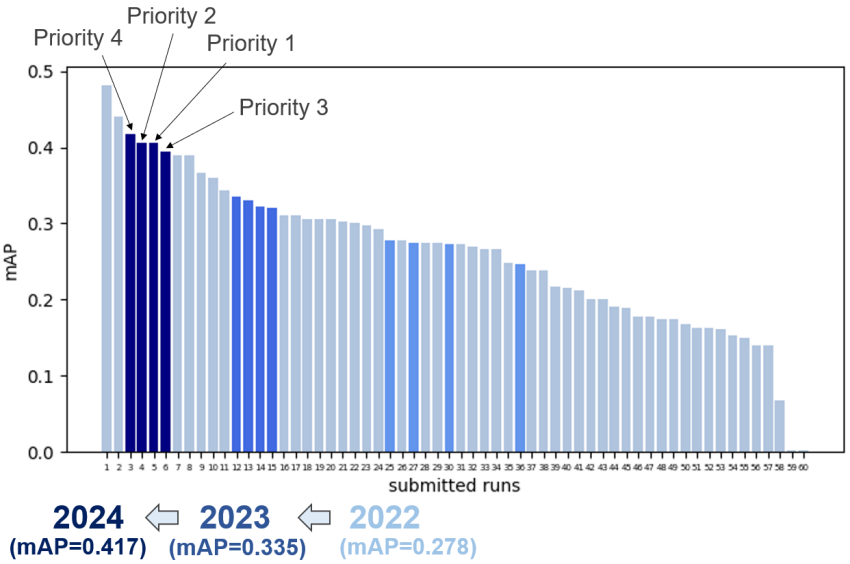

Fig. 3. Progress task results (Set-B). Our team achieved 2nd place, with mAP increasing from 0.278 (2022) to 0.335 (2023) and 0.417 (2024), driven by improved models and image-based query expansion.

In the Progress task, our team achieved second place in both Fig. 2 (Set-A) and Fig. 3 (Set-B).

Over the years, the mean average precision (mAP) has improved significantly from 2022 to 2023 and further to 2024 in both sets. For set-A, mAP increased from 0.256 in 2022 to 0.286 in 2023, and further to 0.351 in 2024. For set-B, mAP improved from 0.278 in 2022 to 0.335 in 2023, and further to 0.417 in 2024. The improvements from 2022 to 2023 can be attributed to leveraging the latest visual-semantic embedding models, which enhanced the baseline accuracy, and expanding the query texts. From 2023 to 2024, the improvements were achieved not only by utilizing state-of-the-art visual-semantic embedding models, but also by incorporating query expansion through image generation from textual queries. This approach contributed to a broader and more diverse search space, resulting in higher overall accuracy.

2 VTT Task

2.1 Overview

This year, our team continued to participate in the VTT task and submitted both the main task and sub-task inferences.

This year's approach to the main task is based on last year's components with some updates. The first update is the variation of the individual finetuned base models. This year, in addition to BLIP2, GIT, and InstructBLIP, which were used last year, we also used LLaVa. (We also used BLIP3;xGen-MM as the base for reranking and refining, although we are not finetuning it.) The second update is prompt tuning of LLM. We used LLM to summarize the captions that are the output of each model, by tuning the prompt used for generation. We also implemented another approach by generating EntityGraph of each caption using LLM, and then referencing the generated Entity-Graphs for summarization. The third update is Data Augmentation. We augmented the data using BackTranslation and similar sentence generation with GPT, and used the augmented data for fine-tuning. This year, we added these updates to last year's components and submitted four run files to the main task. Compared to last year's test data, we have seen improvements of about 0.04 points in CIDEr-D and about 0.03 points in SPICE, based on the indicators we have.

For each subtask, we submitted three files with the highest validation scores among the individual models.

2.2 Methods

Our approach consists of three main components.

The first is fine-tuning. In this process, we fine-tune multiple image/video captioning models on the VTT dataset(V3C). The four base models used were BLIP2, GIT, and InstructBLIP, which were also used last year, as well as the newly adopted LLaVa. In addition, this time we also used data that was augmented during finetuning. Data augmentation involves back translation using Google Translation and similar sentence generation using GPT3.5.

The second is reranking. In this process, the captions generated by the fine-tuned model are compared to the original video, and then sorted by score. We also hope that the Image Captioning model will be able to generate captions for each frame in a video and then rerank them to select captions that will more comprehensively cover the content of the video.

The third is refining. Similar to reranking, we merge captions using captions generated from fine-tuned models. However, while the concept of reranking is to select the best sentence from the generated captions, the concept of refining is to summarize into a single sentence from multiple generated captions, and cover the overall image/video from each caption. In addition to summarizing each caption and correcting grammar, this year the prompts also generate an Entity-Graph for each caption and match entities between captions. We also used the pre-trained model BLIP3(xGen-MM) for reranking and refining, although we were unable to train it on time this time.

We used these approaches for the main task, and for the subtask, we selected and submitted the inferences of the Finetuned models.

BLIP2 BLIP2 [3] is an abbreviation for "Bootstrap vision-language pretraining model." and is a model that connects the vision model and LLM. BLIP2 introduces a Q-former that connects the encoder and decoder, and the learning process is divided into two steps. In the first training step, the parameters of the large language model (LLM) were fixed, and in the second step, the parameters of the image encoder were fixed. Dividing the learning process into two steps helps to bridge the modality gap and reduces the number of parameters trained simultaneously, resulting in better efficiency.

GIT GIT[15], which was proposed by Microsoft, is a lightweight vision language model with high performance in tasks such as captioning and VQA, and is comparable to SOTA method on various major benchmarks. Specifically, GIT is recognized as a model that places significant emphasis on the visual recognition ability. For this contest, we employed the video-compatible version of the model, which is pre-trained on Vatex[**?**] video datasets. This model takes 6 frames of images and describes the content of given image sequences. To feed the model, we sample 6 frames from each video with equal intervals, regardless of the length of the video and output single caption from each video. We fine-tuned this GIT-vatex model with the V3C dataset and updated all parameters.

InstructBLIP InstructBLIP, as introduced in [16], augments the process of extracting visual features and instructions from images and prompts. This is achieved by integrating instructions into not only the frozen LLM, but also into the Query Transformer (QFormer). During the fine-tuning phase of InstructBLIP, we designated the instruction as "Describe." We fed only the instruction into QFormer, whereas both the instruction and ground truth were input into the LLM layer for loss computation. During inference, both QFormer and the LLM were furnished with the same instruction to generate captions. We also tuned the instruction fed into the model, and used the fine-tuned inference as the final run of the InstructBLIP model.

LLaVA LLaVA[19] connects pre-trained CLIP ViT-L/14 visual encoder and large language model Vicuna, using a simple projection matrix. LLaVA is trained to follow the instructions that include both text and visual features. In this contest, we employed LLaVA-1.5-7b model and fine-tuned the model with the fixed prompt "Describe the picture.". We trained the model using DMO[24], which can directly optimize non-differentiable metrics with fewer computational cost than conventional Reinforcement learning. DMO is a variant of offline reinforcement learning and can maximize metrics

such as BLEU[21] and CIDEr[22], which are non-differentiable. During the training, only the parameters of projection layers were updated.

BLIP3(xGen-MM) BLIP3 (xGen-MM)[20] is an extension of BLIP2, which (1) expands it to large and diverse datasets, (2) makes it scalable to image sizes by replacing the Q-Former layer with an embedding that uses ViT[23] image patches, and (3) simplifies the training steps by standardizing the loss across multiple training stages. The pre-trained model xgen-mm-phi3-mini-base-r-v1.5 is used with fixed parameters and is set as the target caption for reranking and refinement. The prompt uses the default "USER: <image>\n What's the content of the image? ASSISTANT:", and about four Few-Shots from V3C are used for inference.

Reranking We generated captions from BLIP2, GIT, InstructBLIP, LLaVA and BLIP3 and calculated the similarity between each caption and the video using vision/text encoders. The caption with the highest similarity was used as the final output for submission. To measure the similarity, eight frames were sampled equally from the video, and the embedding of the image in each frame along with the embedding of the query text were obtained. The final similarity scores were calculated as the cosine similarities between those embeddings. We employed EVA-CLIP[18] as the vision and text encoder to calculate the embeddings.

Refining For the refining, the reranked generated captions from BLIP2, GIT, InstructBLIP, LLaVA and BLIP3 were used to create a single summarized caption using LLM. The intuition behind performing LLM-based summarization is to capture the missing information when compared to captioning by a single model. The custom prompt for the summarization was refactored several times to capture the information across the multiple captions. Apart from performing general summarization based on described features on captions, we also experimented using entity graphs for the summarization task.
For the entity graph based summarization, we first instructed the LLM to generate individual entity graphs for the reranked captions from different VLMs.The generated entity graphs were then provided along with the instruction to generate a single caption, which serves as the final response. From our validation results, we observed that the entity graph based summarization resulted in the better results when compared to general summarization.

2.3 Experiments

The base models, GIT[5], BLIP2[6], InstructBLIP[7], LLaVA[8] and BLIP3[9], use models implemented by Hugging Face as the basis for fine-tuning. These models were fine-tuned using the TV22 training dataset and the model parameters were selected based on the evaluation results of the TV22 test dataset. The input videos were divided into eight frames with a resolution of 224×224. For the image captions, one random frame

[5] https://huggingface.co/microsoft/git-large-vatex
[6] https://huggingface.co/Salesforce/blip2-opt-2.7b
[7] https://huggingface.co/Salesforce/instructblip-vicuna-7b
[8] https://huggingface.co/liuhaotian/llava-v1.5-7b
[9] https://huggingface.co/Salesforce/xgen-mm-phi3-mini-base-r-v1.5

from the set of frames was selected and used for training. For text preprocessing, texts longer than 150 words were excluded from the training data. Additionally, a period was included for sentences without periods. The pretrained tokenizers specific to each model were used for tokenization. The training parameters for each model are listed in Table 2. Pretrained models were used as is for reranking and refining. A refining prompt was created, as mentioned in the Methods section.

Table 2. Hyperparameters for fine-tuning GIT, BLIP2, InstructBLIP, and LLaVA

Hyperparameters	GIT w/ SCST	BLIP2	InstructBLIP	LLaVA
batchsize	256	48	24	256
epochs	5	20	20	5
optimizer	AdamW	AdamW	AdamW	AdamW
learning rate	1e-05	1e-06	1e-04	1e-05
warmup-step	none	1000	none	none
beamsize	1	20	20	1

2.4 Results

Our team's results are shown in Table 3 for the main task and Table 4 for the robustness task. In the main task results, SPICE came in first, METEOR came in second, and the others came in third, as compared to other teams. This year, Runfile3, which was set as Primary, was as expected and seemed to produce better results overall than the other Runfiles. Although they achieved a good score in SPICE, just like last year they were left far behind other teams in CIDEr and other areas, so this remains an area they need to work on. As with the main task, the robustness task also saw their team leave a large gap behind compared to the team in first place among CIDEr.

Table 3. Results of our submitted runs for TREC2024 VTT task.

Runfile	Method	Primary	CIDER	CIDER-D	BLEU	METEOR	SPICE	STS				
								table 1	table 2	table 3	table 4	table 5
1	Reranking		0.801	0.435	**0.1364**	0.4002	0.182	0.487	0.476	0.518	0.481	0.483
2	Refining1		0.789	0.378	0.0943	0.3982	0.174	0.491	0.4483	0.507	0.48	0.485
3	Refining2	√	0.841	0.441	0.1283	0.4067	**0.199**	**0.505**	**0.502**	**0.525**	**0.488**	**0.5**
4	BLIP2 w/pseudo data		**0.848**	**0.523**	0.1354	**0.4092**	0.18	0.485	0.483	0.509	0.485	0.484

Table 4. Robustness results of our submitted runs for TREC2024 VTT task.

Runfile	Method	Primary	CIDER	CIDER-D	BLEU	METEOR	SPICE	STS				
								table 1	table 2	table 3	table 4	table 5
1	GIT		0.761	0.435	0.1145	0.3741	0.150	0.457	0.456	0.474	0.446	0.448
2	BLIP2		**0.831**	**0.510**	**0.1606**	0.4017	0.168	**0.485**	0.475	0.5	0.473	0.478
3	BLIP2 w/pseudo data	√	0.830	0.501	0.1371	**0.4029**	**0.174**	0.482	**0.48**	**0.506**	**0.48**	**0.48**

References

1. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," arXiv:2103.00020, 2021.
2. J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation," arXiv:2201.12086, 2022.
3. J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," arXiv:2301.12597, 2023.
4. X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid Loss for Language Image Pre-Training," In Proc. of IEEE/CVF International Conference on Computer Vision (ICCV), 2023.
5. Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "EVA-CLIP: Improved Training Techniques for CLIP at Scale," arXiv:2303.15389, 2023.
6. H. Xu, S. Xie, X. E. Tan, P.-Y. Huang, R. Howes, V. Sharma, S.-W. Li, G. Ghosh, L. Zettlemoyer, and C. Feichtenhofer, "Demystifying CLIP Data," arXiv:2309.16671, 2023.
7. C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5B: An open large-scale dataset for training next generation image-text models," In Proc of the 36th Conference on Neural Information Processing Systems (NeurIPS), 2022.
8. C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision," In Proc. of the International Conference on Machine Learning (ICML), 2021.
9. B. Zhang, P. Zhang, X. Dong, Y. Zang, and J. Wang, "Long-CLIP: Unlocking the Long-Text Capability of CLIP," In Proc. of the European Conference on Computer Vision (ECCV), 2024.
10. Z. Lai, H. Zhang, B. Zhang, W. Wu, H. Bai, A. Timofeev, X. Du, Z. Gan, J. Shan, C.-N. Chuah, Y. Yang, and M. Cao, "VeCLIP: Improving CLIP Training via Visual-enriched Captions," arXiv:2310.07699, 2023.
11. J. Chen, Q. Yu, X. Shen, A. Yuille, and L.-C. Chen, "ViTamin: Designing Scalable Vision Models in the Vision-Language Era," In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.
12. H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual Instruction Tuning," In Proc. of the Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS), 2023.
13. Marah Abdin, et al., "Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone," arXiv:2404.14219 , 2024.
14. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," In Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.10684–10695, 2022.
15. J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, "GIT: A Generative Image-to-text Transformer for Vision and Language," arXiv:2205.14100, 2022.
16. H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," arXiv:2304.08485, 2023.
17. S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical Sequence Training for Image Captioning," arXiv:1612.00563, 2017.
18. Y. Fang, Q. Sun, X. Wang, T. Huang, X. Wang, and Y. Cao, "EVA-02: A Visual Representation for Neon Genesis," arXiv:2303.11331, 2023.
19. H. Liu, C. Li, Y. Li, YJ. Lee, "Improved Baselines with Visual Instruction Tuning," arxiv:2310.03744, 2024
20. L. Xue, M. Shu, A. Awadalla, J. Wang, A. Yan, S. Purushwalkam, H. Zhou, V. Prabhu, Y. Dai, et al., "xGen-MM (BLIP-3): A Family of Open Large Multimodal Models," arxiv2408.08872, 2024

21. K. Papineni, S. Roukos, T. Ward, WJ. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002
22. R. Vedantam, C. Lawrence Zitnick, D. Parikh, “Cider: Consensus-based image description evaluation,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
23. A. Dosovitskiy, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” arXiv2010.11929, 2021
24. T. Takumi and S. Yuma and T. Hiroki and T. Hayato and S. Haruki and K. Aiswariya and N. Hiroki and H. Takayuki and U. Kazuya, “Direct Metric Optimization for Image Captioning through Reward-Weighted Augmented Data Utilization,” Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), 2024

A Appendix A

A.1 VTT main task

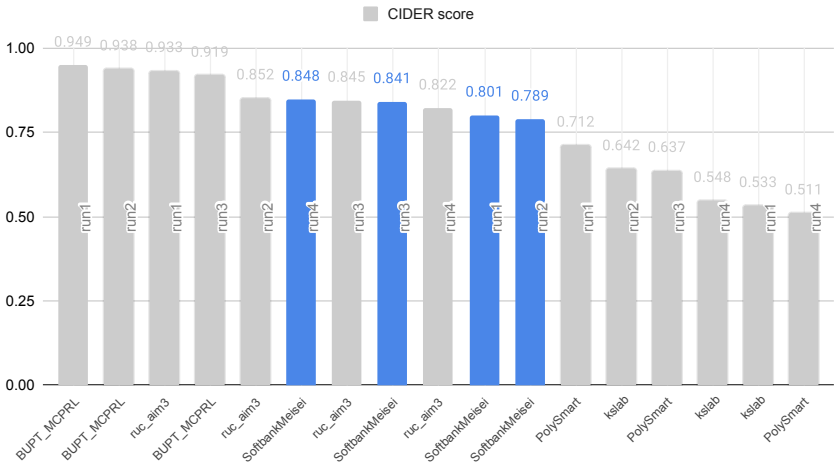

Fig. 4. Main task results CIDEr

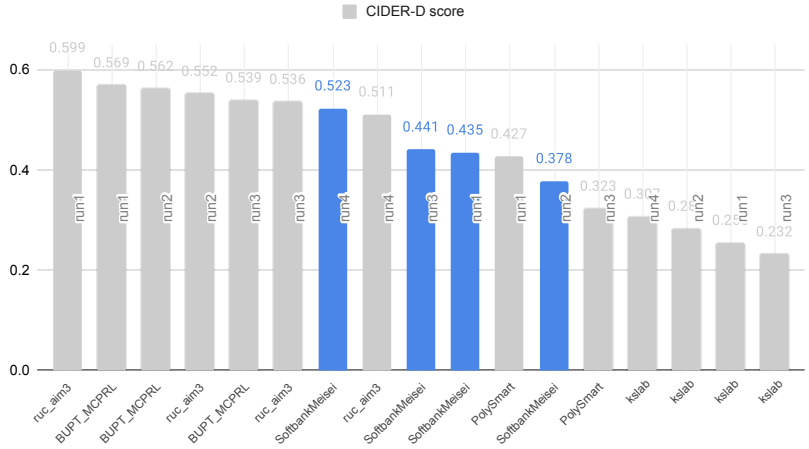

Fig. 5. Main task results CIDEr-D

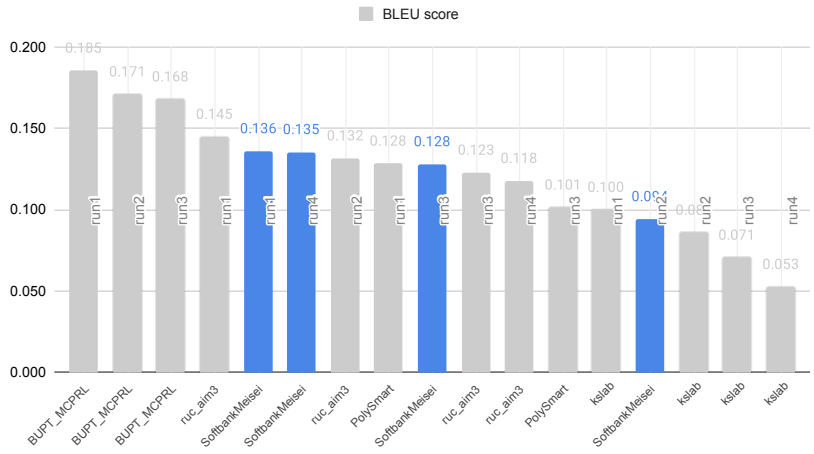

Fig. 6. Main task results BLEU

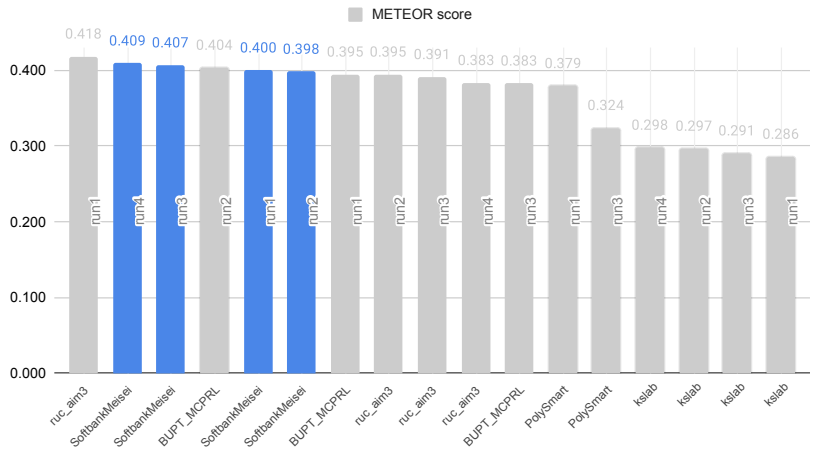

Fig. 7. Main task results METEOR

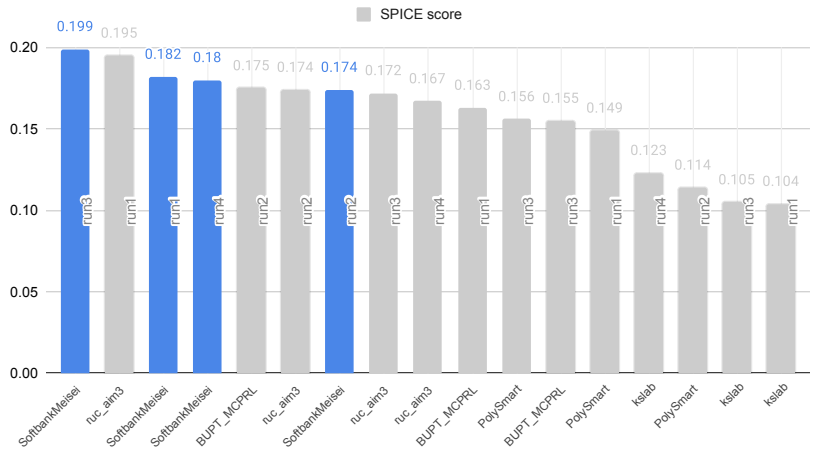

Fig. 8. main task results SPICE

A.2 VTT robustness task

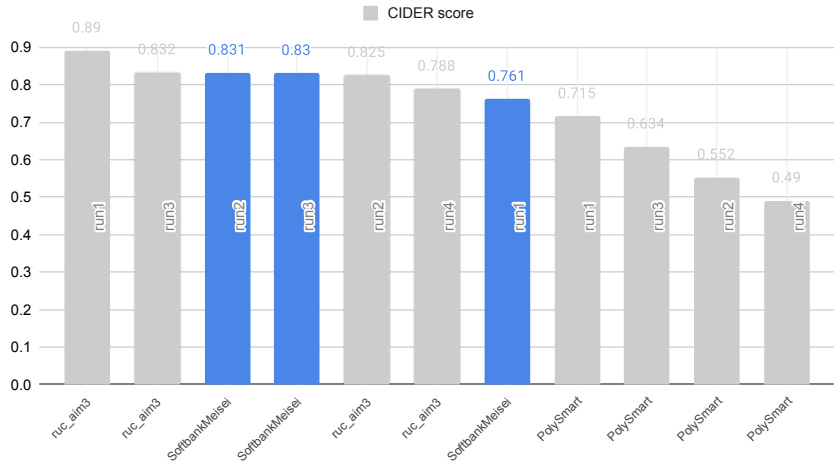

Fig. 9. Robustness task results CIDEr

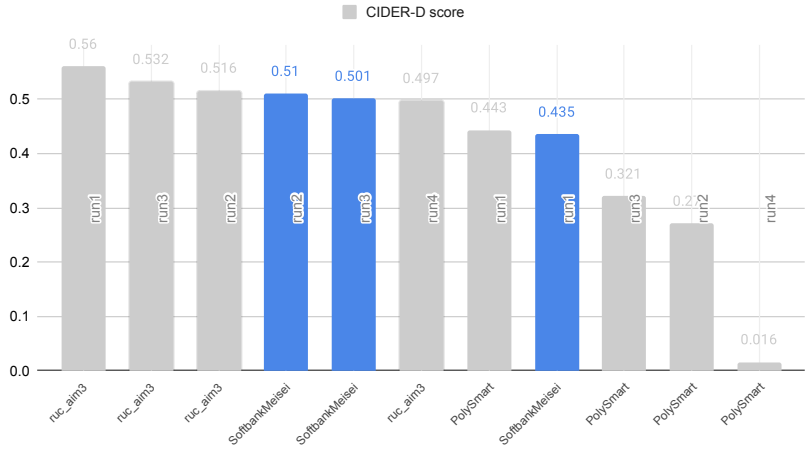

Fig. 10. Robustness task results CIDEr-D

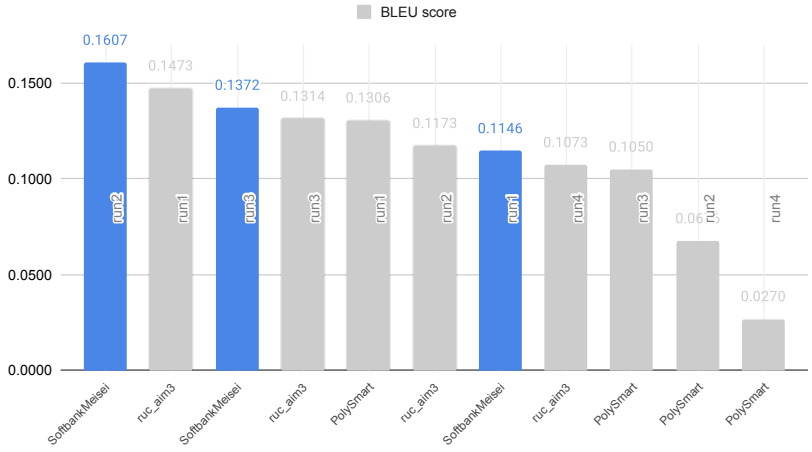

Fig. 11. Robustness task results BLEU

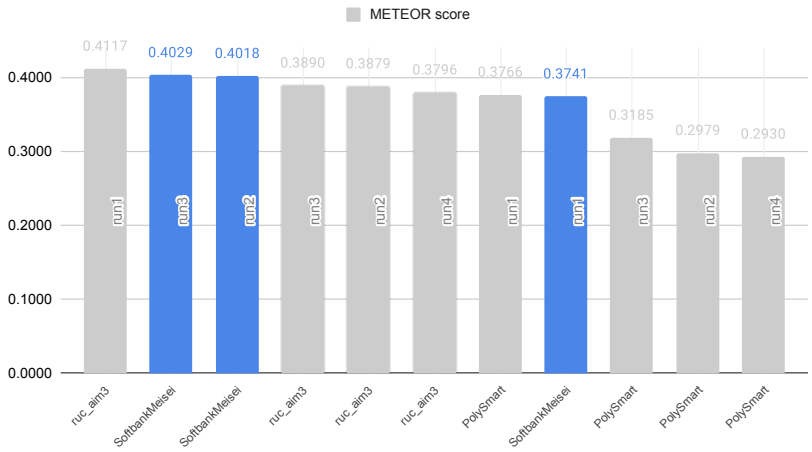

Fig. 12. Robustness task results METEOR

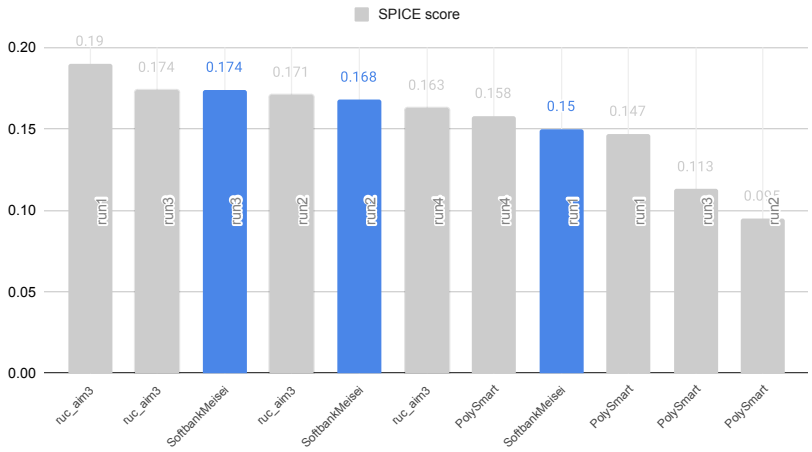

Fig. 13. Robustness task results SPICE