Waseda_Meisei_SoftBank at TRECVID 2023

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

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

[2] Faculty of Science and Engineering, Waseda University,
Room 40-701, Waseda-machi 27, Shinjuku-ku, Tokyo 162–0042, Japan

[3] AI Architect Department Technical Planning & Development Division,
Service Planning Technology Division, SoftBank Corporation
1-7-1 Kaigan, Minato-ku, Tokyo 105–7529, Japan

[4] Global Information and Telecommunication Institute, Waseda University,
Room 55-208, Okubo 3-4-1, Shinjuku-ku, Tokyo 162–0042, Japan

kazuya.ueki@meisei-u.ac.jp

Abstract. The Waseda_Meisei_SoftBank team participated in the ad-hoc video search (AVS) and video-to-text (VTT) tasks at TRECVID 2023 [1]. In the AVS task of this year, we continued to employ the visual-semantic embedding approach, submitting four fully automatic systems for both the main and progress tasks, following the methodology of last year. The best-performing system achieved an mAP of 0.285 for the main task, securing the second position among the participating teams. During the progress task, it obtained an mAP of 0.286, attaining the highest overall position. In addition, our team participated in the VTT task again this year. We also participated in the subtask that was included this year. Our system for this year consisted of multiple caption models and three components for re-ranking and refining the generated sentences. In the main task, it obtained BLEU score of 0.108, METEOR of 0.335, and SPICE of 0.152, achieving the top overall position.

1 AVS Task

1.1 System Overview

In the past, we constructed systems that combined concept-based approaches, where concepts such as words and phrases were pretrained and combined with visual-semantic embedding techniques. However, in recent years, there has been a significant advancement in the technology of training on large-scale image datasets with captions, resulting in improved accuracy. As a result, this year, we exclusively employed visual-semantic embedding models to build our system, following the approach from the previous year. For our embedding techniques, we incorporated several methods, including improved visual-semantic embeddings (VSE++) [2], a graph-structured matching network (GSMN) [3], contrastive language-image pre-training (CLIP) [4], self-supervision meets language-image pretraining (SLIP) [5], and the diffusion model. Notably, we endeavored to enhance the image retrieval accuracy by incorporating multiple pretrained models provided by OpenCLIP[5].

[5] https://github.com/mlfoundations/open_clip

1.2 New Initiatives Undertaken This Year

The system update of this year involved not only incorporating newly available high-performance pretrained models, but also experimenting with query expansion using ChatGPT. Assuming that a richer variety of input queries results in an increased number of retrievable images, we explored the generation of alternative expressions of the original query that carry the same meaning but differ in phrasing. Although there are various approaches to altering the original text, we opted to use ChatGPT, which is a sophisticated natural language processing and dialogue generation tool. We attempted to input the following five prompts into ChatGPT:

1. Give me 10 sentences that mean exactly the same as “original query” with slight changes.
2. Give me 10 examples of “original query” that means exactly the same thing, but with a slight change in the sentence.
3. List 10 sentences that mean the same as “original query” with slight modifications.
4. List 10 examples that mean the same thing as “original query,” but with a slight change in the sentence.
5. Give me 10 sentences that mean the same as the following sentence with a slight change of wording: “original query”

For each query, we input the five aforementioned prompts twice, resulting in 100 new sentences. From this set, we employed non-duplicate sentences as novel queries for the image-retrieval process. The aim of this approach was to enhance the diversity and effectiveness of the image retrieval process.

1.3 Integration Approach for Multiple Embedding Methods

We developed our video retrieval systems by calculating the similarity between the textual features of the query sentences and the visual features extracted from the frame images of videos obtained from each embedding model. This was followed by weighted integration of the similarity values from multiple models. Because each model varies in its training methodology and the dataset on which it has been trained, the complementarity among these models typically improves the overall accuracy when their results are combined. Given that some models outperform others, we adjusted the fusion weights based on the ground truth from the previous year to optimize the mean average precision. However, overfitting to the ground truth of the previous year could lead to reduced generalization performance and potentially lower accuracy for the queries of this year. Therefore, we developed two systems: one that uses a hard fusion weight determined by the previous ground truth and another that employs a softer, more evenly distributed fusion weight.

1.4 Submissions and Results

This year, we created four different automatic systems and submitted their results. The distinctions among these systems lie in the approach used for integrating the models: determining hard weights, setting soft weights, and the inclusion or exclusion of query expansion using ChatGPT. The performances of the systems submitted this year are presented in Table 1. Regarding the fusion weights, the systems that were assigned hard weights exhibited better accuracy than those that were assigned soft weights.

Table 1. Our submitted systems for TRECVID 2023.

System	Fusion weight	Query expansion (ChatGPT)	mAP	
			Main task	Progress task
1	Soft	√	0.269	0.272
2	Hard	√	**0.285**	**0.286**
3	Soft		0.270	0.269
4	Hard		0.281	0.283

Consequently, it became evident that the superiority of the models had relatively little variation depending on the input queries, underscoring the significance of prioritizing the selection and utilization of high-accuracy models. However, the effect of the query expansion using ChatGPT was relatively modest. The preliminary experiments involving the expansion of queries from the previous year yielded improvements, suggesting that the effectiveness of the query expansion may vary depending on the query type. In the future, we plan to conduct further analyses to understand the conditions under which the accuracy improves and to refine our approach accordingly.

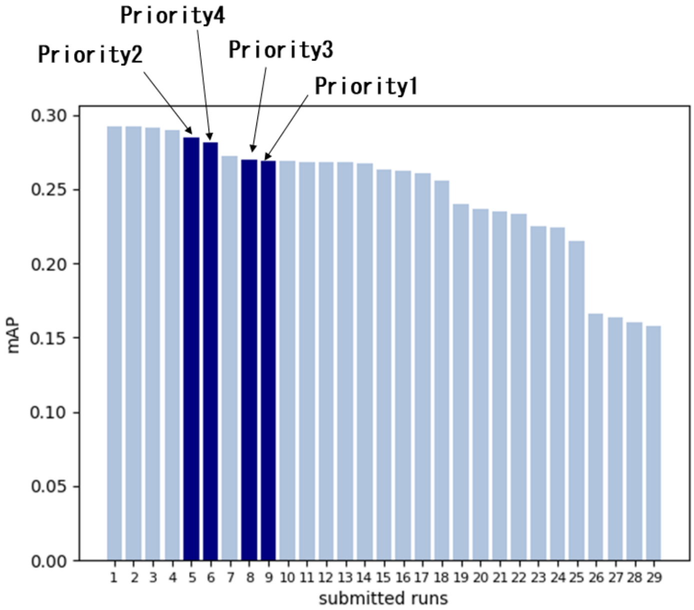

Fig. 1. Results of all fully automatic systems for all teams that submitted to the main task.

The results for all teams that submitted to the main task are shown in Fig. 1. The four systems that we submitted achieved rankings of 5th, 6th, 8th, and 9th among all systems submitted by participating teams, resulting in 2nd place for our team in the team-specific standings.

The results of all systems submitted to the progress task are presented in Fig. 2. Our four systems that were submitted this year secured the 1st to 4th positions, exhibiting improved accuracy compared to the systems that we submitted last year. This increase in accuracy is attributed to the influence of the newly introduced models

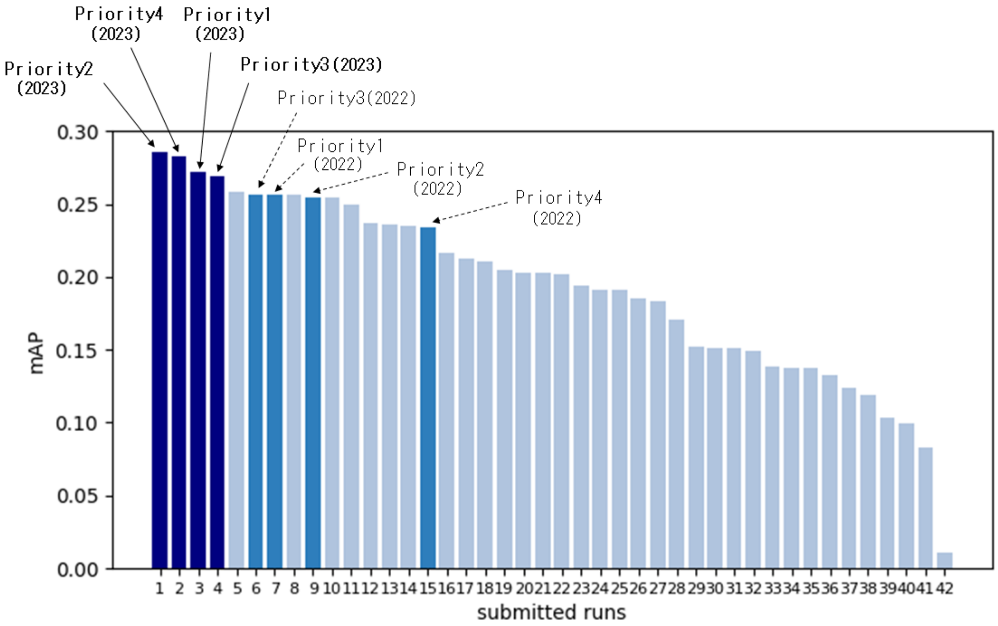

Fig. 2. Results of all fully automatic systems for all teams that submitted to the progress task.

this year and the query expansion using ChatGPT. In future, we plan to investigate the contributions of each approach further.

2 VTT Task

2.1 Overview

Our team participated in the VTT task again this year following our participation in the previous year. We also participated in an additional subtask that was introduced this year. Based on the results of the previous year, we developed two main strategies for the approach this year. The first involves the use of large vision language models. With recent advancements, such as the introduction of ChatGPT, the number of model parameters has increased significantly. This trend is no exception in vision language models, as large and highly accurate models have emerged. Therefore, it is essential to leverage these models to achieve optimal performance. Furthermore, as mentioned in our strategy from the previous year, the video-captioning dataset is relatively small. To address this issue, we aimed to acquire generalization capabilities efficiently using pretrained models. The second strategy involves merging the results generated by multiple models. Last year, I worked alone as the developer. However, this year, we had a team of three members, which allowed us to conduct more experiments. We focused on merging the individual results to combine the strengths of each model and to improve the overall performance.

Following the aforementioned strategy, we submitted a system comprising three components for the competition this year: fine-tuning, reranking, and refining. Among the publicly released results for the main task, the proposed system achieved the highest scores for BLEU, METEOR, and SPICE. The scores for these metrics were 0.108, 0.335, and 0.152, respectively.

2.2 Methods

Our approach consists of three main components. The first is fine-tuning. We fine-tuned the three image/video caption models using the VTT dataset. The base models selected were BLIP2, GIT, and InstructBLIP. By fine-tuning these models, we aimed to improve their performance, specifically for the VTT task. The second component is reranking. We calculated the similarity between the captions generated by the fine-tuned models and the videos on which they were based. The captions were ranked based on their scores. Because we used image caption models, this approach allowed us to assess how well each model captured all situations within the videos and selected the best captions. The third component is refining. We performed caption merging using the captions generated by the fine-tuned models. This involved summarizing, making grammatical corrections, and controlling the length of the captions. We used the same approach for both the main task and the subtask in this competition.

BLIP2 BLIP2 [13] is an abbreviation for "Bootstrap vision-language pretraining model." It is the successor model to BLIP, which was used by several teams last year. 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 [12], which was proposed by Microsoft, is a powerful vision language model with high performance in tasks such as captioning and VQA, and is comparable to SOTA method on various major benchmarks. For this contest, we fine-tuned the model with the V3C dataset and updated all parameters. In addition to fine-tuning, we further trained the model using SCST [15], which can directly optimize non-differentiable metrics. SCST is a method of maximizing scores by applying a reinforcement learning algorithm to metrics such as BLEU and CIDEr, which are non-differentiable. As SCST can calculate rewards only after generating the entire caption, it generates very long computational graphs. Thus, it can be applied only to lightweight models such as GIT. During training, the vision model part was frozen and only the language model part was updated.

InstructBLIP InstructBLIP, as introduced in [14], 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.

Reranking We generated captions from BLIP2, GIT, and InstructBLIP and calculated the similarity between each caption and the video using a CLIP-like vision text encoder. 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[16] as the vision text encoder to calculate the embeddings.

Refining For this process, we used the GPT3.5 model developed by OpenAI (we would have liked to use GPT4, but it was not available in time for our preparations). Prompts were developed to incorporate the Who, Where, What, and When information that is also described in the annotation rules.

- First step:
  identify descriptions that are not found in other captions
  → color information, person/background information
  consider synonyms and ungenerated words
  summary under 30 words

- Second step:
  revised text with emphasis on readability

2.3 Experiments

The base models, GIT[6], BLIP2[7], and InstructBLIP[8], 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 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 Table2. 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, and InstructBLIP

Hyperparameters	GIT w/ SCST	BLIP2	InstructBLIP
batchsize	2	48	48
epochs	30	20	20
optimizer	AdamW	AdamW	AdamW
learning rate	1e-05	1e-06	1e-04
warmup-step	none	1000	none
beamsize	-	20	20

[6] `https://huggingface.co/microsoft/git-large-vatex`
[7] `https://huggingface.co/Salesforce/blip2-opt-2.7b`
[8] `https://huggingface.co/Salesforce/instructblip-vicuna-7b`

2.4 Results

Our results for the main task and robustness task are presented in Tables 3 and 4, respectively.

Table 3. Results of our submitted runs for TRECVID 2023 VTT task.

Runfile	Method	Primary	CIDER	CIDER-D	BLEU	METEOR	SPICE	STS				
								table 1	table 2	table 3	table 4	table 5
1	GIT		0.628	0.287	0.096	0.315	0.126	0.449	0.458	0.463	0.458	0.447
2	BLIP2	√	**0.682**	0.324	**0.108**	0.324	0.133	0.451	0.459	0.465	0.457	0.453
3	Reranking		0.642	0.320	0.079	0.331	**0.152**	**0.472**	**0.474**	**0.478**	**0.477**	**0.472**
4	Refining		0.673	**0.348**	0.103	**0.335**	0.143	0.460	0.466	0.472	0.468	0.461

Table 4. Robustness results of our submitted runs for TRECVID 2023 VTT task.

Runfile	Method	Primary	CIDER	CIDER-D	BLEU	METEOR	SPICE	STS				
								table 1	table 2	table 3	table 4	table 5
1	GIT		0.630	0.286	0.092	0.313	0.122	0.183	0.189	0.192	0.194	0.185
2	BLIP2	√	**0.677**	0.313	**0.105**	0.323	0.132	0.188	0.191	0.193	0.195	0.187
3	Reranking		0.637	0.313	0.078	0.3311	**0.151**	**0.219**	**0.225**	**0.230**	**0.225**	**0.220**
4	Refining		0.677	**0.340**	0.097	**0.3314**	0.141	0.204	0.211	0.213	0.213	0.206

We achieved the highest scores for the three metrics in the main task. However, there was a significant gap compared to the other teams in CIDER, which places a strong emphasis on captions. We would like to address this as a challenge for next year. Additionally, although our attempts at data augmentation did not work well, we are actively working on improving it in the upcoming year.

Acknowledgments

This work was partially supported by the Telecommunications Advancement Foundation.

References

1. G. Awad, K. Curtis, A. A. Butt, J. Fiscus, A. Godil, Y. Lee, A. Delgado, E. Godard, B. Chocot, L. Diduch, Y. Graham, and G. Quénot, "TRECVID 2023 - A series of evaluation tracks in video understanding," In Proceedings of TRECVID 2023, 2023.
2. F. Faghri, D. J. Fleet, R. Kiros, and S. Fidler, "VSE++: Improved Visual-Semantic Embeddings," arXiv:1707.05612, 2017.
3. C. Liu, Z. Mao, T. Zhang, H. Xie, B. Wang, and Y. Zhang, "Graph Structured Network for Image-Text Matching," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
4. 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.
5. N. Mu, A. Kirillov, D. Wagner, and S. Xie, "SLIP: Self-supervision meets Language-Image Pre-training," arXiv:2112.12750, 2021.

6. C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, “Collecting Image Annotations Using Amazon’s Mechanical Turk,” Proceedings of the NAACLHLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, pp.139–147, 2010.
7. P. Young, A. Lai, M. Hodosh and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” Transactions of the Association for Computational Linguistics. vol. 2, pp. 67–78, 2014.
8. T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” arXiv:1405.0312, 2014.
9. P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning,” Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 2556–2565, 2018.
10. R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalanditis, L.-J. Li, D.A. Shamma, M.S. Bernstein, L. Fei-Fei, Y. Kalantidis, L.-J. Li, D.A. Shamma, M.S. Bernstein, and F.-F. Li, “Visual Genome: Connecting language and vision using crowdsourced dense image annotations,” arXiv:1602.07332, 2016.
11. J. Xu, T. Mei, T. Yao, and Y. Rui, “MSR-VTT: A Large Video Description Dataset for Bridging Video and Language,” In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
12. 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.
13. 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.
14. H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” arXiv:2304.08485, 2023.
15. S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical Sequence Training for Image Captioning,” arXiv:1612.00563, 2017.
16. 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.