Kindai University, Osaka Gakuin University and Osaka University at TRECVID 2021 AVS Task

Kimiaki Shirahama*, Takumi Sato*, Norihiro Yamawaki*,
Takashi Matsubara† and Kuniaki Uehara‡
* Department of Informatics, Kindai University
† Graduate School of Engineering Science, Osaka University
‡ Department of Business Administration, Osaka Gakuin University
Contact: shirahama@info.kindai.ac.jp

***Abstract*—This paper presents our method developed for Ad-hoc Video Search (AVS) task in TRECVID 2021. Our method is built using Conceptual Captions dataset [1], MS COCO dataset [2], MSR-VTT dataset [3] and TGIF dataset [4] (models pre-trained on a dataset of 940 million social media images with 1500 noisy hashtags [5], ImageNet dataset [6], Wall Street Journal part of Penn Treebank (PTB) dataset [7] and Visual Genome [17] are internally used). Our method performs retrieval by fusing three component models, VSE++ that uses image features to match a shot with a topic [8], dual encoder that uses image and video features for shot-topic matching [9], and Stacked Cross Attention Network (SCAN) that uses image features of regions in a shot [10]. Specially, our five submitted runs are characterised by the following combinations of component models:**

1) **F_M_C_D_kindai_ogu_osaka.21_1: VSE++, dual encoder, and SCAN that matches regions in a shot with words and phrases in a topic by extracting the constituency tree of the topic.**
2) **F_M_C_D_kindai_ogu_osaka.21_2: VSE++, dual encoder, and SCAN that carries out a slightly different matching of regions with words and phrases.**
3) **F_M_C_D_kindai_ogu_osaka.21_3: VSE++, dual encoder and SCAN that matches regions only with words (our baseline).**
4) **F_M_C_D_kindai_ogu_osaka.21_4: The same combination to F_M_C_D_kindai_ogu_osaka.21_2 except that the fusion is based on normalised scores from the component models.**
5) **F_M_N_D_kindai_ogu_osaka.21_5: SCAN that matches regions with words and phrases to find unique shots.**

The evaluation results show that the MAPs of F_M_C_D_kindai_ogu_osaka.21_1 at the main and progressive tasks are 0.213 and 0.271 respectively, while the corresponding MAPs of F_M_C_D_kindai_ogu_osaka.21_3 are 0.190 and 0.264. This validates the effectiveness of fine-grained matching of regions with words and phrases based on the linguistic structure of a topic. Finally, F_M_C_D_kindai_ogu_osaka.21_1 achieved the top AP for topic 666 "a man wearing a blue jacket", which suggests that fine-grained matching could be especially useful for topics involving nouns with modifiers. This will be further explored in our future work.

I. Introduction

We are continuously participating in TRECVID for objective performance comparison between our system and systems developed all over the world [11]. This year we participated in Ad-hoc Video Search (AVS) [12] to examine the effectiveness of fine-grained matching between visual features of a shot and textural features of a topic by considering the linguistic structure of the topic. This is inspired by our past experiences to apply Stacked Cross Attention Network (SCAN) [10] to AVS task in TRECVID 2019 and 2020 [13], [14]. Roughly speaking, SCAN was used to compute the relevance of a shot to a topic by matching regions in the shot with words in the topic. However, this matching does not fit human's perception because he/she checks not only whether regions corresponding to words exist in the shot, but also whether those regions suit to phrases involving multiple words. In addition, the lack of considering phrases causes to retrieve several false positive shots where words in one phrase are matched with different regions. Taking "red dress" as an example, such a false positive shot includes the region of a dress and the region of a red object (e.g., red ball, red light, fire, bricks etc.). To realise shot-topic matching akin to human perception and reduce false positive shots, we extend SCAN to SCAN_tree that can perform fine-grained matching between regions in a shot and words/phrases in a topic by extracting the constituency tree of the topic.

II. Our AVS Method

Our AVS method consists of three component models, VSE++ [8], dual encoder [9] and SCAN_tree. This section presents these component models by putting a special focus on SCAN_tree. Different combinations of component models to define the submitted runs will be described in the next section.

A. VSE++

VSE++ is a simple but effective visual-semantic embedding model that maps visual and textual features into a common space, so that the relevance of each shot to a topic can be computed [8]. VSE++ consists of an image encoder that extracts a visual feature from an image, a text encoder that extracts a textual feature from a caption, and Fully-Connected (FC) layers that map the visual and textual features into a common space. A pre-trained model is usually used as the image encoder. In particular, VSE++ in our method utilises ResNeXt-101 WSL (32x48d) [5] that is pre-trained in weakly-supervised fashion on 940M social media images with 1.5K noisy hashtags and fine-tuned using ImageNet dataset [6]. The text encoder is implemented using a network consisting of a word embedding layer followed by a layer of Gated Recurrent

Unit (GRU). Given a training dataset, the text encoder and FC layers are optimised so that the visual feature of an image and the textual feature of the corresponding caption are projected close to each other in the common space. In addition, the optimisation aims the projection where the projected feature of an image (or a caption) is distant from the projected features of irrelevant captions (or images).

VSE++ in our method is trained on the dataset created by combining 3M image-caption pairs in Conceptual Captions (CC) dataset [1] and 0.6M pairs in MS COCO dataset [2]. For encoding visual features in a shot, the image encoder is applied to 10 equidistantly-sampled frames together with the keyframe, because analysing multiple frames in a shot usually leads to a performance improvement [14]. Average-pooling is then employed to aggregate features extracted from these 11 frames into a single vector, which is subsequently projected into the common space by the trained VSE++. Also, a topic is encoded into a textual feature that is then projected into the common space. Finally, the cosine similarity between the aggregated visual feature of the shot and the textual feature of the topic in the common space is used as the relevance of the shot to the topic. Please refer to our notebook papers in 2019 and 2020 for more details about VSE++ [13], [14].

B. Dual Encoder

Dual encoder is a visual-semantic embedding model that performs the following three-level encodings for both a shot and a topic [9]: The first-level extracts an overall feature, specifically, the average of visual features extracted from frames that are sampled every 0.5 seconds from a shot, and the average of one-hot vectors representing each word in a topic. The second-level encoding extracts a feature reflecting sequential relations. A sequence of visual features extracted above is fed into a bi-directional GRU, and the average of hidden states computed for these features is regarded as the feature obtained by the second-level encoding. Similarly, the aforementioned one-hot vectors are fed into a word embedding to obtain a sequence of textual features, and then a bi-directional GRU is employed to obtain the average of hidden states. The third-level encoding extracts a feature representing short-term sequential relations. For both shot and topic sides, convolutions characterised by different kernel sizes are applied to the sequence of hidden states obtained at the second-level encoding. Afterwards, max-pooling is used to summarise features obtained by the convolution with one kernel size into a single vector, and the concatenation of vectors from all kernel sizes is treated as the output of the third-level encoding. Finally, the concatenation of features from all levels in the shot side and the one in the topic side are projected into a common space.

Dual encoder in our method is trained using MSR-VTT dataset [3] and TGIF dataset [4]. Differently from the original implementation [9], we extracted visual features of a frame as the concatenation of features obtained using two pre-trained models, ResNeXt-101 WSL (32x48d) [5] and vision transformer pre-trained on ImageNet dataset [15].

C. SCAN_tree

SCAN_tree consists of two main processes, the extraction of a constituency tree for each topic and the matching of regions with words and phrases represented by the constituency tree. The parser developed in [7] is used for our constituency tree extraction. Roughly speaking, the parser starts with an empty tree and sequentially adds each word to the tree by performing the action to attach the word as a child node of an existing node or the action to juxtapose the word as a sibling to an existing node by creating a shared parent node. A neural network to conduct this action selection is trained on Wall Street Journal part of Penn Treebank (PTB) dataset [7]. Fig. 1 shows an example of constituency tree extracted for topic 668 "a person wearing an apron indoors". As shown in this figure, nodes in the tree is labelled with constituent tags like DT (Determiner), NN (Noun), NP (Noun Phrase) and so on.

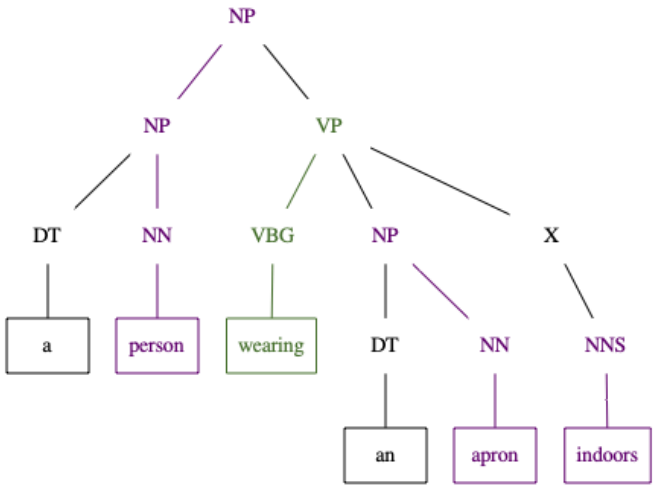

Fig. 1. An examle of constituency tree extracted for topic 668 "a person wearing an apron indoors".

Let us consider to match regions in an image with words and phrases contained in the constituency tree of a caption. To this end, regions are encoded into features in the following way: First, the image is analysed to extract 36 salient regions each of which is likely to include a concept with an attribute, such as "blue water", "black hair" or "floral dress". Such salient regions are extracted using a bottom-up attention model [16] that is implemented with Faster R-CNN based on ResNet101 backbone and trained on Visual Genome dataset [17]. Then, each region is represented by a 2048-dimensional feature that is the output of the intermediate layer (pool5_falt) of the bottom-up attention model, and transformed into a 1024-dimensional feature via an FC layer.

Words and phrases are encoded using a Tree-LSTM that is an extended LSTM to propagate hidden states and memory cells based on the topology of a tree [18]. In particular, a Child-Sum Tree-LSTM is used to perform the following bottom-up propagation: Given a constituency tree for a caption, words corresponding to leaf nodes are firstly encoded

into 300-dimensional vectors via a word embedding layer. These vectors are then used to compute 1024-dimensional hidden states and memory cells for leaf nodes. The process for each internal node starts with defining the "overall" hidden state from its child nodes as the sum of their hidden states. The overall hidden state is used to compute values of the input, output and forget gates. Here, a forget gate value is separately calculated for each child node to signify whether it is strongly related to the node or not. Afterwords, a hidden state and a memory cell for the node are obtained using the gate values. This way, hidden states and memory cells are propagated from leaf nodes to the root node corresponding to the whole of the caption. Note that no external input like word embedding features exist for internal nodes. In other words, the propagation for each internal node is based only on hidden states from its child nodes. The hidden state of each node is regarded as the feature of the corresponding word or phrase.

For simplicity, words and phrases are collectively called "tokens" as long as there is no need to distinguish between them. Regions and tokens are matched using the attention mechanism of the original SCAN [10]. Roughly speaking, an attention between a region and a token is computed as their normalised similarity lying between 0 and 1. That is, the attention represents a probabilistic relevance of matching the region with the token. Then, the "token-level" relevance of how suitable regions in the image are for the token, is computed as the cosine similarity between the token's feature and the average of regions' features weighted by their attentions to the token. Finally, the "caption-level" relevance of the image is measured as the average of token-level relevances over all tokens. In this framework, the FC layer in the region encoder, and the word embedding layer and the Tree-LSTM in the token encoder are optimised so that the caption-level relevances are high and low for relevant image-caption pairs and irrelevant ones, respectively. This optimisation consequently leads to semantically meaningful matching between regions and tokens. For more details, please refer to the original SCAN paper [10] and our notebook paper in 2019 [13].

Since all tokens in a caption are not necessary for examining the relevance of an image, the following four variants of SCAN_tree are devised by changing types of tokens to be matched with regions. The first variant considers the set of 21 token types including NN (Noun), NP (Noun Phrase), VB (Verb), VP (Verb Phrase), JJ (Adjective), ADJP (Adjective Phrase), RBR (Adverb), ADVP (Adverb Phrase), CD (Cardinal number) and so on. The second variant reduces this set to the one of 13 token types that are related to nouns and verbs. The third variant uses the further reduced set of 6 types related only to nouns. The last variant is the original SCAN that takes no consideration of phrases and uses all the words in a caption. Although token selection is theoretically crucial for accurate retrieval, our preliminary experiments showed that the performances of all the four variants are similar. But, one notable thing is that different shots are retrieved by different variants. Considering this diversity in retrieved shots, the final retrieval result of SCAN_tree is acquired by late fusion where the final relevance of each shot to a topic is computed as the average of caption-level relevances obtained by the four variants. Our preliminary experiments validated that late fusion significantly boosts the retrieval performance.

III. Results

Our submitted five runs are configured by combining VSE++, dual encoder and SCAN_tree as follows:

1) F_M_C_D_kindai_ogu_osaka.21_1: The overall relevance of a test shot to a topic is computed just by summing their similarity in the common space obtained with VSE++, their similarity in the common space with dual encoder, and the relevance with SCAN_tree.
2) F_M_C_D_kindai_ogu_osaka.21_2: The only difference from the first run is in SCAN_tree's Tree-LSTM. While the first run uses word embedding features for leaf nodes, this run attempts to sophisticate each of these features by considering its surrounding features based on a bi-direction GRU. That is, the hidden state obtained for each word is used as the external input of the corresponding leaf node in the Tree-LSTM.
3) F_M_C_D_kindai_ogu_osaka.21_3: The overall relevance of a test shot to a topic is computed by summing their similarity in the common space with VSE++, their similarity in the common space with dual encoder, and the relevance computed with the original SCAN. That is, this is our baseline run to examine the effectiveness of considering phrases in the topic by SCAN_tree.
4) F_M_C_D_kindai_ogu_osaka.21_4: This run is the same to the second run except that the similarity between a test shot and a topic by VSE++, the one by dual encoder, and the relevance by SCAN_tree are linearly normalised to have the minimum 0 and the maximum 1 before they are summed up.
5) F_M_N_D_kindai_ogu_osaka.21_5: We believe that few teams carries out fine-grained matching between regions and token, so this novelty run only uses SCAN_tree to find unique shots that are not found by other teams.

Fig. 2 shows the ranking list of the runs submitted to the main AVS task. As depicted by the five red-coloured bars at the right, our submitted runs are unfortunately ranked at the bottom. However, the MAPs of F_M_C_D_kindai_ogu_osaka.21_1 (0.213), F_M_C_D_kindai_ogu_osaka.21_2 (0.193) and F_M_C_D_kindai_ogu_osaka.21_4 (0.199) are higher than the MAP of the baseline F_M_C_D_kindai_ogu_osaka.21_3 (0.190). This validates the effectiveness of SCAN_tree that considers phrases in a topic together with words.

Fig. 3 displays the ranking list of the runs submitted to the progress task. This figure presents a different trend from Fig. 2. Specifically, in Fig. 3, our submitted runs are not ranked at the bottom compared to the runs that are submitted in 2021. In particular, F_M_C_D_kindai_ogu_osaka.21_1 whose MAP is 0.271 is ranked at about the middle among those runs. This suggests that the performance of our method significantly depends on topics, and the analysis of such a dependence is

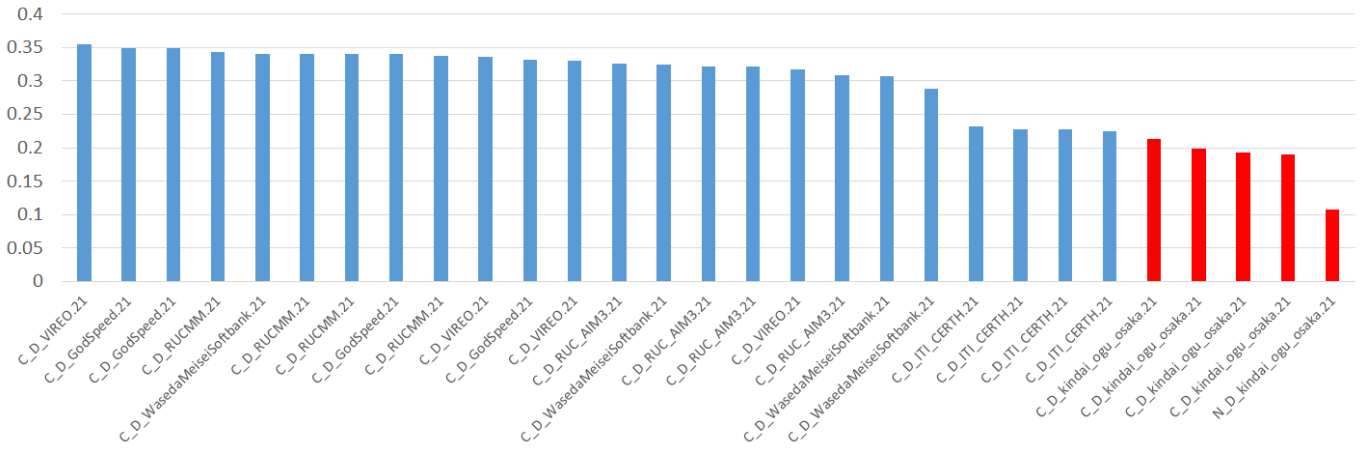

Fig. 2. Ranking list of the runs submitted to for the main AVS task.

one of our future work. In addition, the comparison among the yellow-, green- and red-coloured bars that respectively correspond to our submitted runs in 2019, 2020 and 2021, clearly shows the performance improvements that we have achieved in the three years.

Finally, Fig. 4 exhibits the comparison of region-token matching between SCAN_tree (especially the second variant considering tokens related to nouns and verbs) and the original SCAN for topic 616 "A woman wearing a red dress outside in the daytime". In this figure, each rectangle drawn on the frame presents the most attentive region for each token. Fig. 4 (b) shows inconsistent region-word matching where the most attentive region for "woman" is different from such regions for "red" and "dress", although this woman is wearing the red dress. Compared to this, Fig. 4 (a) demonstrates that SCAN_tree can select suitable tokens for examining the relevance of the frame and put attention on semantically appropriate regions.

IV. Conclusion and Future Work

This paper introduced our method developed for TRECVID 2021 AVS task. It combines three component models, VSE++, dual encoder and SCAN_tree. Especially, SCAN_tree is devised by extending SCAN to perform matching between regions in a frame and words and phrases in a topic by extracting the constituency tree of the topic. The evaluation results validates the effectiveness of SCAN_tree. Our future work to further improve SCAN_tree includes the adoption of a more advanced bottom-up attention model to extract a better feature of a region, the replacement of a Tree-LSTM with a transformer, and the consideration of relations among regions to restrict region-token matching.

References

[1] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proc. of ACL 2018*, 2018, p. 2556–2565.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. of ECCV 2014*, 2014, pp. 740–755.

[3] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proc. of CVPR 2016*, 2016, pp. 5288–5296.

[4] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo, "Tgif: A new dataset and benchmark on animated gif description," in *Proc. of CVPR 2016*, 2016, pp. 4641–4650.

[5] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proc. of ECCV 2018*, 2018, pp. 185–201.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[7] K. Yang and J. Deng, "Strongly incremental constituency parsing with graph neural networks," in *Proc. of NeurIPS 2020*, 2020.

[8] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," in *Proc. of BMVC 2018*, 2018.

[9] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang, "Dual encoding for zero-example video retrieval," in *Proc. of CVPR 2019*, 2019, pp. 9338–9347.

[10] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc. of ECCV 2018*, 2018, pp. 212–228.

[11] G. Awad, A. A. Butt, K. Curtis, J. Fiscus, A. Godil, Y. Lee, A. Delgado, J. Zhang, E. Godard, B. Chocot, L. Diduch, J. Liu, Y. Graham, G. J. F. Jones, , and G. Quénot, "Evaluating multiple video understanding and retrieval tasks at trecvid 2021," in *Proceedings of TRECVID 2021*. NIST, USA, 2021.

[12] J. Lokoč, W. Bailer, K. Schoeffmann, B. Muenzer, and G. Awad, "On influential trends in interactive video retrieval: Video browser showdown 2015–2017," *IEEE Transactions on Multimedia*, vol. 20, no. 12, p. 3361–3376, 2018.

[13] K. Shirahama, D. Sakurai, M. Takashi, and K. Uehara, "Kindai university and kobe university at TRECVID 2019 avs task," in *Proc. of TRECVID 2019*, 2019.

[14] D. Mukai, R. Utsunomiya, S. Utsuki, K. Shirahama, M. Takashi, and K. Uehara, "Kindai university and osaka gakuin university at trecvid 2020 avs and actev tasks," in *Proc. of TRECVID 2020*, 2020.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *Proc. of ICLR 2021*, 2021.

[16] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. of CVPR 2018*, 2018, pp. 6077–6086.

[17] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced

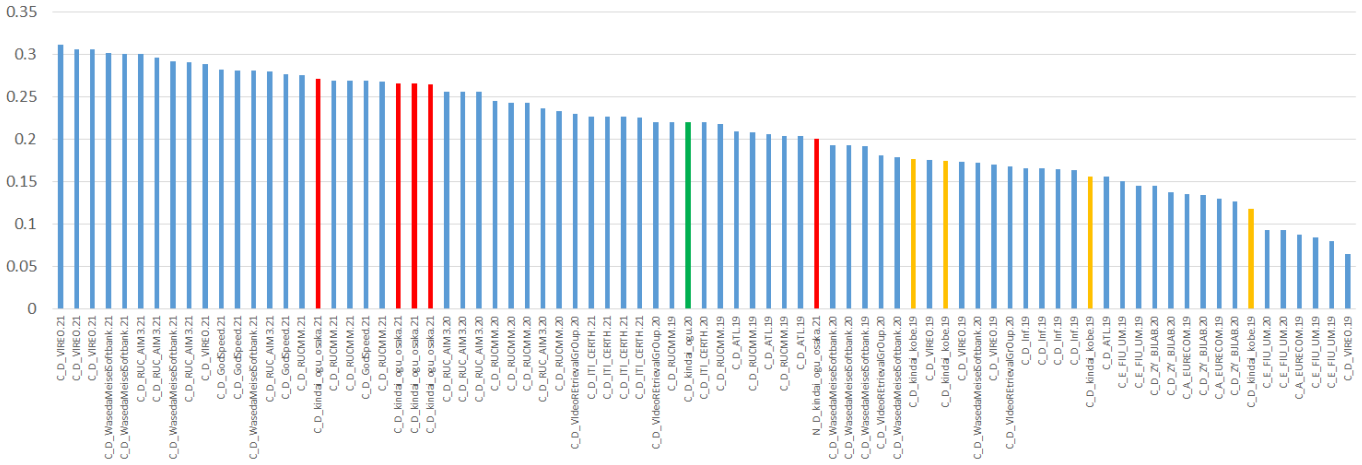

Fig. 3. Ranking list of the runs submitted to the progress AVS task.

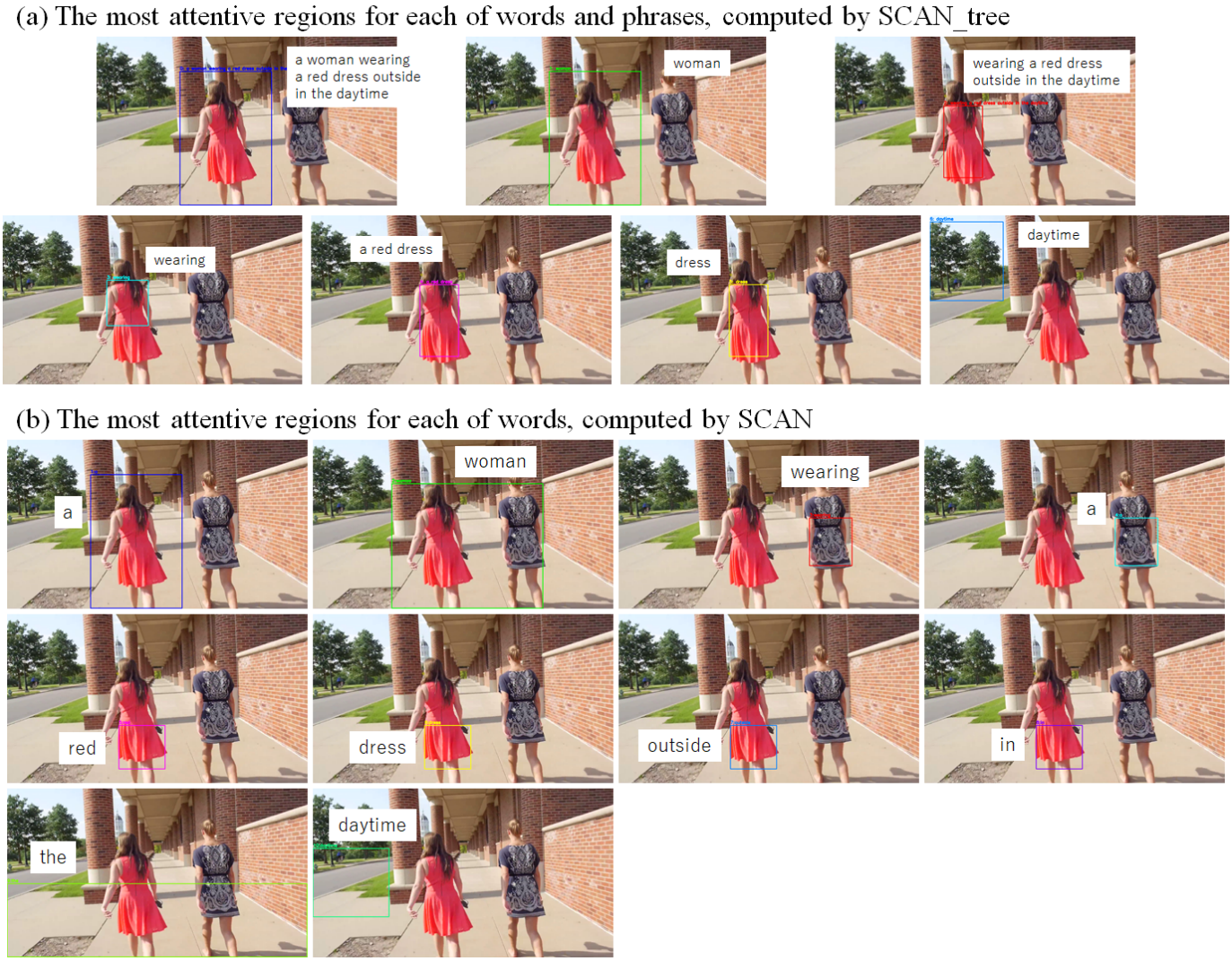

Fig. 4. Comparison of region-token matching between SCAN_tree and SCAN for topic 616 “A woman wearing a red dress outside in the daytime”.

dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[18] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. of ACL 2015*, 2015, pp. 1556–1566.