Waseda_Meisei_SoftBank at TRECVID 2021: Ad-hoc Video Search

Kazuya Ueki[1,2], Takayuki Hori[3,4], Yongbeom Kim[3], and Yuma Suzuki[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 Solution Division, 5G & IoT Solution 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 TRECVID 2021 ad-hoc video search (AVS) task. This year, as last year, we submitted manually assisted and fully automatic runs for both the main task and the progress subtask. Our approach consisted of concept-based video retrieval and visual-semantic embedding. We used a visual-semantic embedding approach for the fully automatic runs and a fusion of both concept-based and visual-semantic embedding approaches for the manually assisted runs. Our best fully automatic run for the main task achieved a mean average precision (mAP) of 34.1%, which ranked fourth among all participants. Our best manually assisted run for the main task achieved an mAP of 33.1%, which ranked second among all manually assisted systems.

1 System Description

We submitted both fully automatic and manually assisted systems to the Text Retrieval Conference Video Retrieval Evaluation (TRECVID) 2021 ad-hoc video search (AVS) task [1]. This section introduces how both systems were created.

1.1 Fully Automatic Systems

The fully-automatic systems were created by integrating various visual-semantic embedding approaches. Last year, the systems were developed using only improved visual-semantic embeddings (VSE++) [2] as the embedding method. This year, recently proposed embedding methods were introduced, such as a graph-structured matching network (GSMN) [3], contrastive language-image pre-training (CLIP) [4], and object-semantics aligned pre-training (Oscar) [5].

VSE++ can extract global representations of images and text, but cannot determine the relationship between objects in an image and words in a sentence. Therefore, we introduced a GSMN, which can model objects, relationships, and attributes as structured phrases through node- and structure-level correspondences. CLIP, proposed by OpenAI, has become a hot topic in the field of image and text retrieval since the beginning of 2021. CLIP achieves zero-shot, highly accurate image retrieval without

fine-tuning by pre-training on a large number of text/image pairs (over 400 million). Oscar uses object tags detected in the image as anchor points, which greatly improves the generalizability of the pre-trained models.

For each of the embedding approaches presented above, we computed the scores for all test video shots. The following three types of video-shot frames were used in each approach, depending on when the work was done and how fast the calculations were performed:

- $Frame_k$: Use only key frames
- $Frame_{10}$: Use the middle 10 frames of the video divided into 11 equal parts
- $Frame_{e10}$: Use every 10 frames.

In the following, we explain how each model was created and how the scores for each video shot were calculated.

1. VSE++

   We used the implementation of VSE++[5] for training. To train the visual-semantic embedding, four image-caption datasets, Flickr8k [6], Flickr30k [7], MS-COCO [8], and Conceptual Captions [9], were used. The total number of image captions was 3,428,009, including 40,000 from Flickr8k, 155,070 from Flickr30k, 423,915 from MS-COCO, and 2,809,024 from Conceptual Captions[6]. We used a gated recurrent unit (GRU) for feature extraction from query sentences and the *ResNet-50*, *ResNet-101*, and *ResNet-152* models for feature extraction from images.
   Because of the large amount of training data, 500,000 training data pairs and 50,000 validation data pairs were randomly selected to train the visual-semantic embedding models. We repeated this data-selection process 32 times for each of the three types of ResNet model, and trained 96 embedding models. For the score calculation, we obtained 192 different scores for each shot using both $Frame_{10}$ and $Frame_{e10}$ for the 96 models. After adding all 192 scores, the final scores were obtained by min-max normalization; that is, the maximum and minimum scores were 1.0 and 0.0, respectively.

2. GSMN

   The visual features of the GSMN were extracted using the bottom-up attention model[7] and the pre-trained bottom-up attention model provided there. The bottom-up attention model is based on training *Faster R-CNN* with *ResNet-101*, using object and attribute annotations from the Visual Genome [10]. GRU was used to extract features from the text.
   To train the GSMN models, we used the GSMN implementation[8] and a total of 3,755,503 image-text pairs, including 40,000 from Flickr8k, 155,070 from Flickr30k, 596,435 from MS-COCO, 2,809,018 from Conceptual Captions, and 154,980 from MSR-VTT [11]. Because of the large amount of training data, we divided the training data and created nine models[9]. For the score calculation, we obtained nine

[5] `https://github.com/fartashf/vsepp`

[6] The total amount of data in the Conceptual Captions dataset was 3,334,173, including 3,318,333 training data and 15,840 validation data; however, only 2,809,024 downloadable data were used.

[7] `https://github.com/peteanderson80/bottom-up-attention`

[8] `https://github.com/CrossmodalGroup/GSMN`

[9] The data were divided into ten parts and models were created; however, one of them failed to be created, so nine models were finally used.

different scores for each shot using only $Frame_{e10}$. The final scores were calculated by min-max normalization after adding all scores, as in the case of VSE++.

3. CLIP

   We did not train the models ourselves, but used the pre-trained models provided in the CLIP implementation[10]. We used four types of pre-trained CLIP models: *ViT-B/32*, *RN50*, *RN101*, and *RN50x4*. *ViT-B/32* is based on a vision-transformer architecture. *RN50* and *RN101* are architectures equivalent to *ResNet-50* and *ResNet-101*, respectively. *RN50x4* is an *RN50* scaled up $4\times\times$, according to the EfficientNet scaling rule. For the score calculation, we obtained eight different scores for each shot by using both $Frame_{10}$ and $Frame_{e10}$ for the four models. The final scores were calculated by min-max normalization after adding all the scores, as in the case of VSE++ and GSMN.

4. Oscar
   Similar to CLIP, we did not train any models for Oscar, but used the large pre-trained Oscar model available on GitHub[11]. For the score calculation, we obtained only one score for each shot, using $Frame_k$ for the large pre-trained Oscar model. The final scores were calculated in the same manner as for the other embedding methods.

In this year's automatic systems, the test data were ranked according to the scores, which were calculated by simply adding the scores from the four different embedding methods, multiplied by the fusion weights. The fusion weights were determined manually by evaluating the 2019 and 2020 TRECVID AVS tasks. This year, we submitted four fully automatic runs (Automatic1, Automatic2, Automatic3, and Automatic4). The fusion weights of VSE++, GSMN, CLIP, and Oscar used in our systems are as follows:

- Automatic1: 5 : 5 : 10 : 1
- Automatic2: 3 : 3 : 10 : 1
- Automatic3: 7 : 7 : 10 : 1
- Automatic4: 10 : 10 : 10 : 1.

The reason for the high fusion weights of the CLIP models is that VSE++ and GSMN use models that were trained using the same training data (Flickr8k, Flickr30k, MS-COCO, and Conceptual Captions), whereas CLIP uses a model that was trained on 400 million pairs, which is different from the training data used for VSE++ and GSMN. In Oscar, the fusion ratio was set to low because only one model was used, and the score was calculated on only one keyframe, $Frame_k$, from each video.

1.2 Manually Assisted Systems

The manually assisted systems were created by combining the concept-based method and the visual-semantic embedding methods, because we found, from the results of the past few years, that the concept-based and visual-semantic embedding approaches

[10] `https://github.com/openai/CLIP`

[11] `https://github.com/microsoft/Oscar/blob/master/MODEL_ZOO.md#Image-Text-Retrieval`

Table 1. Concept bank used in our systems.

Name	Database	# Concepts	Concept Type(s)	Models
TRECVID346	TRECVID SIN [12]	346	Person, Object, Scene, Action	GoogLeNet + SVM
FCVID239	FCVID [13]	239	Person, Object, Scene, Action	GoogLeNet + SVM
UCF101	UCF101 [14]	101	Action	GoogLeNet + SVM
PLACES205	Places [15]	205	Scene	AlexNet
PLACES365	Places	365	Scene	GoogLeNet
HYBRID1183	Places, ImageNet [16]	1,183	Person, Object, Scene	AlexNet
IMAGENET1000	ImageNet	1,000	Person, Object	GoogLeNet
IMAGENET4000	ImageNet	4,000	Person, Object	GoogLeNet
IMAGENET4437	ImageNet	4,437	Person, Object	GoogLeNet
IMAGENET8201	ImageNet	8,201	Person, Object	GoogLeNet
IMAGENET12988	ImageNet	12,988	Person, Object	GoogLeNet
IMAGENET21841	ImageNet	21,841	Person, Object	GoogLeNet
ACTIVITYNET200	ActivityNet [17]	200	Action	GoogLeNet + SVM
KINETICS400	Kinetics [18]	400	Action	3D-ResNet
ATTRIBUTES300	Visual Genome [10]	300	Attributes of persons/objects	GoogLeNet + SVM
RELATIONSHIPS53	Visual Genome	53	Relationships b/w persons/objects	GoogLeNet + SVM
FACES40	CelebA [19]	40	Face Attributes	face detector + CNN

are complementary. The visual-semantic embedding methods used in the manually assisted systems are the same as those used in the fully automatic systems. For the concept-based approach, we used a large concept bank, comprised of several concept types, as shown in Table 1. It contains classifiers, such as persons, objects, scenes, and actions, to support the various forms of query sentences. Using this concept bank, all concept scores for all videos were calculated. Because the concept bank used this year is exactly the same as the one used last year, we omit the explanation of how we created the concept bank and refer to last year's notebook paper for details [20].

After calculating the concept scores[12] for every video sequence in advance, we retrieved videos using word-based keyword selection through the following pipeline.

1. Extract one or more keywords from a query sentence.
2. Select one or more concept classifiers related to a keyword. The corresponding concept may not exist in the concept bank.
3. For each video, calculate the score for the query sentence by integrating the scores from multiple concept classifiers.

Given a query sentence, we manually selected some visually important keywords. For example, given the query sentence, "two or more ducks swimming in a pond," we selected the keywords "duck," "swimming," and "pond." We then matched the keywords with concepts, using a concept classifier. Semantically similar concepts were also chosen using the word2vec algorithm [21] to select as many concept classifiers as possible. The advantage of the concept-based method is that it can accurately extract the videos corresponding to words, such as "duck," "swimming," and "pond." However, it has the disadvantage that phrases like "two or more" and "in a pond" are ignored.

As in the past few years, the visual-semantic embedding and concept-based approaches were combined to re-rank the video-retrieval result using reciprocal rank fusion (RRF) [22],

$$RRF_{score} = \sum_{r \in R} \frac{1}{k + r}, \tag{1}$$

where R is the set of rankings and k is a fixed parameter.

[12] The score for each semantic concept was normalized for all test-shot iterations using a min-max normalization; that is, the maximum and minimum scores were 1.0 (most probable) and 0.0 (least probable), respectively.

2 Submission Results

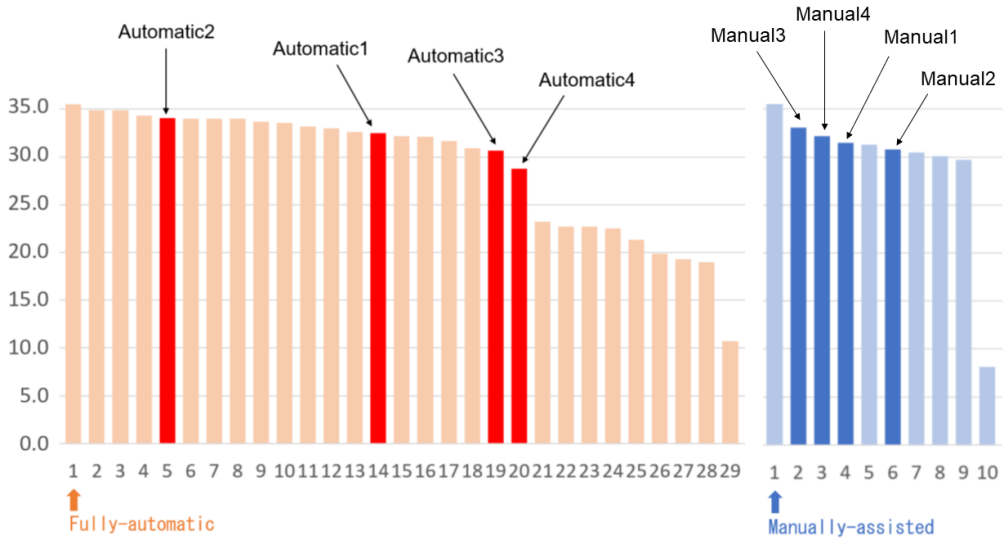

Fig. 1. Results of all the systems for all teams that submitted to the main task in the 2021 submission, including both fully automatic (red) and manually assisted (blue).

This year, we submitted four fully automatic runs (Automatic1, Automatic2, Automatic3, and Automatic4) and four manually assisted runs (Manual1, Manual2, Manual3, and Manual4) to both the main task and the progress subtask.

The results for all teams that submitted to the main task are shown in Fig. 1. In the main task, our best automatic run and manually assisted systems ranked fourth and second, respectively, among all participating teams.

The results for all the fully automatic and manually assisted systems submitted to the progress subtask, which continuously evaluates the systems submitted from 2019 to 2021, are shown in Figs. 2 and 3, respectively. Our best fully automatic system came in second place among all participating teams, and the difference in mean average precision (mAP) between ours and the system with the highest accuracy was only 1.0 (31.1 vs. 30.1). Our best manually assisted system ranked the highest among all the manually assisted systems in three years.

Figures 2 and 3 also show a comparison of the accuracy of the systems we submitted for 2020 and 2021. The main difference between the 2020 and 2021 systems is the embedding method used. The 2020 systems used only VSE++, while the 2021 systems additionally introduced the latest embedding methods GSMN, CLIP, and Oscar. From these differences in accuracy, we can see the progress of embedding methods in the last few years.

The results of our submitted runs for 2021 are listed in Table 2. First, looking at the different fusion weights of the fully automatic runs, we can see that the accuracy is highest when the fusion weight of CLIP is large. This shows that CLIP has a different output tendency and higher retrieval accuracy than VSE++ and GSMN.

For this year's manual systems, we decided to integrate concept-based and visual-semantic embedding approaches, based on the results of the previous year. This is be-

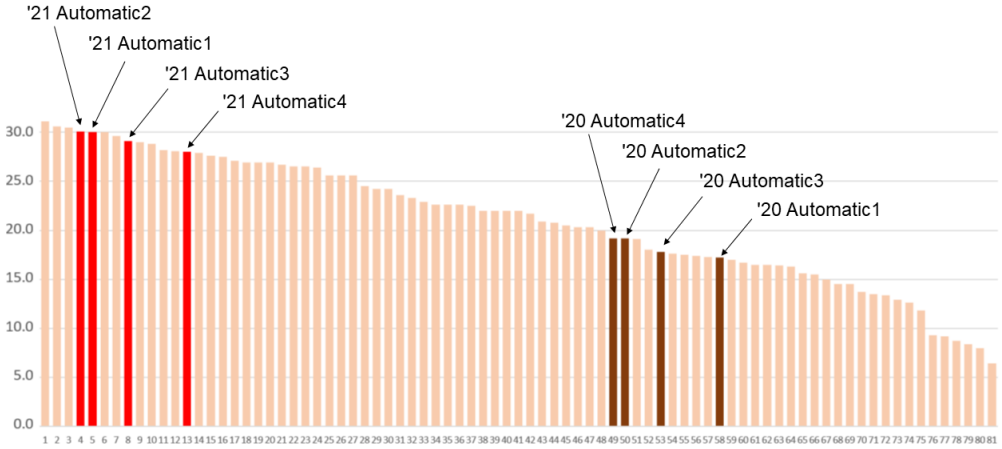

Fig. 2. Results of all the fully automatic systems for all teams that submitted to the progress task in the 2019–2021 submissions.

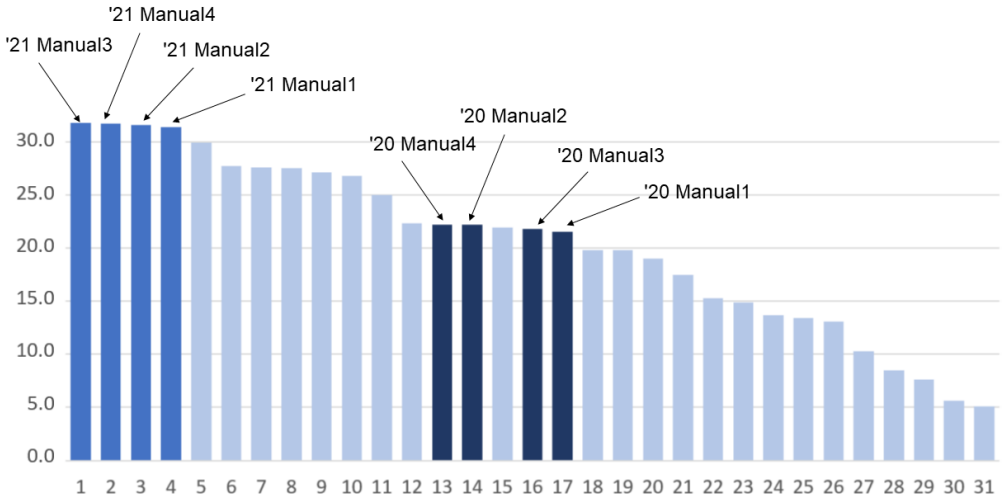

Fig. 3. Results of all the manually assisted systems for all teams that submitted to the progress task in the 2019–2021 submissions.

Table 2. Our submitted runs for TRECVID 2021.

Run name	Fusion weights				Fusion weights		mAP	
	VSE++	GSMN	CLIP	Oscar	embedding	concept	Main	Progress
Automatic1	5	5	10	1	—		32.5	30.0
Automatic2	3	3	10	1	—		**34.1**	30.1
Automatic3	7	7	10	1	—		30.7	29.1
Automatic4	10	10	10	1	—		28.8	28.0
Manual1	5	5	10	1	3	1	31.5	31.4
Manual2	5	5	10	1	2	1	30.8	31.6
Manual3	3	3	10	1	3	1	33.1	**31.8**
Manual4	3	3	10	1	2	1	32.2	31.7

cause the concept-based and embedding methods were complementary, and the video-retrieval accuracy could be improved by integrating them. However, the advantages of the concept-based approach are diminishing, as embedding methods have been greatly improved by the introduction of the newly proposed CLIP and other methods. For example, for the main task, the embedding method alone was better than the fusion of the concept-based and embedding methods. On the other hand, for the progress task, the fusion of the concept-based and embedding methods was better than the embedding method alone. This shows that the performance depends on the query sentences.

3 Conclusion

In the systems submitted this year, we introduced new embedding methods that have been proposed in recent years, such as GSMN, CLIP, and Oscar. The evaluation results showed that the accuracy of the system was significantly better than that of the previous year's system, indicating that the recent pre-training mechanism using large-scale image-text pairs is beneficial.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number 18K11362.

References

1. 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, G. Quénot, "Evaluating Multiple Video Understanding and Retrieval Tasks at TRECVID 2021," In Proc. of TRECVID 2021, 2021.
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, Y. Zhang, "Graph Structured Network for Image-Text Matching," In Proc. of 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, I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," arXiv:2103.00020, 2021.

5. X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, J. Gao, "Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks." In Proc. of European Conference on Computer Vision (ECCV), 2020.
6. C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting Image Annotations Using Amazon's Mechanical Turk," Proc. 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," Proc. 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, Y. Rui, "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language," In Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
12. G. Awad, C. G. M. Snoek, A. F. Smeaton, and G. Quénot, "TRECVid Semantic Indexing of Video: A 6-Year Retrospective," ITE Trans. on MTA vol.4, no.3, pp.187–208, 2016.
13. Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, S.-F. Chang, "Exploiting Feature and Class Relationships in Video Categorization with Regularized Deep Neural Networks," arXiv:1502.07209, 2015.
14. K. Soomro, A. R. Zamir, M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," arXiv:1212.0402, 2012.
15. B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," Advances in Neural Information Processing Systems (NIPS), 2014.
16. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," In Proc. of IEEE Computer Vision and Pattern Recognition (CVPR), 2009.
17. F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles "ActivityNet: A large-scale video benchmark for human activity understanding," In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015), pp.961–970, 2015.
18. W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T.r Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," arXiv: 1705.06950, 2017.
19. Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," In Proc. of International Conference on Computer Vision (ICCV), 2015.
20. K. Ueki, R. Mutou, T. Hori, Y. Kim, Y. Suzuki "Waseda_Meisei_SoftBank at TRECVID 2020: Ad-hoc Video Search," In Proc. of TRECVID 2020, 2020.
21. T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient estimation of word representations in vector space," arXiv:1301.3781, 2013.
22. G. V. Cormack, C. L. Clarke, and S. Buettcher, "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods," Proc. of the 32nd International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp.758–759, 2009.