Unsupervised Context Distillation from Weakly Supervised Data to Augment Video Question Answering [*]

Paul Gaynor

The University of the West Indies,
paul.gaynor@uwimona.edu.jm

Abstract. Anomaly detection by tracking if the context of a video stream has changed could be useful, but supervised training to classify video context can be cumbersome and error prone. Instead, we apply a cascade of clustering techniques that operate on a weakly supervised video data lake to extract a context representation of a video sequence. We then train a bi-directional LSTM model to mimic the functionality of the cascade and predict a context representation from video. Additional experiments have shown that if the context is fed as an additional input to a legacy Video Question Answering solution, loss improves by more than 20% relative to it's baseline after training over 120 epochs, which is significant as current state of the art accuracy for VideoQA solutions is close to 50%. This report is also a demonstration of how to chart a path to freedom from the requirement to explicitly label data, while preserving semantics.

Keywords: clustering, video question answering

1 Introduction

Video stream monitoring relies on a large number of intermediate devices that run many processes [1]. A real time implementation that pushes processing to the edge tries to selectively extract interesting events[2], and then apply additional processing to those events or anomalies. Anomaly detection in video streams is therefore interesting, but is a computationally hard problem[3]. Current methods require significant compute or memory resources over a long period of time. We propose an anomaly detection solution that requires significantly less resources, by classifying the context of the video stream, to which we will refer as the mood, and then simply reporting if the mood has changed. An example of a ***mood change*** is if a video stream shows persons sitting and reading, then later shows persons suddenly running. As there is no known dataset that collates video moods, this work relies on a combination of unsupervised techniques to cluster similar videos, (so each cluster is then considered a mood). We then train a CNN

[*] This work is supported by the Universal Service Fund (Jamaica)

C. Kiran Mai et al. (eds.), *Proceedings of the Fourth International Conference on Advances in Computer Engineering and Communication Systems (ICACECS 2023)*, Atlantis Highlights in Computer Sciences 18,
https://doi.org/10.2991/978-94-6463-314-6_2

based model to mimic the mood generation and assess the matching distributions using the MSRVTT-QA dataset[4]. Further work has demonstrated that if the mood is used as an additional input to a VideoQA system[4], it's accuracy improves, as supported by subset distillation theory[5–7]. Given that each mood is simply a representation of videos which are similar in high dimensional space, it's human readable label is a summarized aggregate of the labels of the video files included in the cluster. As such, examples of labels are

```
1. Who jumps off a roof to talk as they fly through air
2. What model competition where beauty girl walking
3. Who handshakes a commando and handles a revolver
```

The rest of this work is organized as follows: Section 2 presents supporting work, with the method presented in section 3. Experiment setup and results are described in sections 4 and 5. Section 6 presents a discussion, with future work and a conclusion given in sections 7 and 8.

2 Supporting Work

Phenomenal improvements in the classification of environmental patterns have been achieved over the past two decades. Some tasks remain challenging however, such as video anomaly detection [3] as the events sought represent outliers that may occur infrequently, or not, over extended periods of monitoring. Recently proposed solutions such as building and managing a context graph[8], which could become memory intesive, posing the problem as a gaussian expectation-maximization issue[9], which increases computation, or tracking trajectories[10] are likely to cost resources. The expense will be compounded in a long running surveillance application, and is a motivator for this work.

Public release of datasets like Imagenet[11] and the corresponding availability of high performance computing devices spawned a race for top classification performance in multiple domains. AlexNet's victory[12] a decade ago, using a supervised learning technique that applied a convolutional neural network, signalled that the time was ripe for machine learning to leap from the pages of publications into live applications. Unsurprisingly, the next few years witnessed further ground-breaking performances by models like VGG[13], Inception[14] and Resnet[15]. For a short time after, data classification in more complex formats like video, audio and time sequences proved a worthy challenge, until techniques like RNNs[16] and LSTMs[17] became commonplace, along with the attention[18] strategies that drive transformers. Currently, transformer powered language summarization models like BERT[19] are readily available, giving developers options for application integration. As an example while the `pegasus/x-sum` model[20] achieves state of the art performance, licensing concerns may indicate that a less accurate, but more available model like `pegasus/cnn-daily-mail` could be fit for purpose.

Automating question answering over videos (VideoQA), is another currently challenging task, as the accuracy of cutting edge solutions is below 50%[22, 23] The Video Question Answering task is an indirect descendant of the Question Answering challenges[21]. A commonly implemented solution is trained by embedding questions from the training set, or relationships in a knowledge base, into a latent space, and associating answers with each embedding[24, 25]. At inference time an embedding is generated for a posed question, so when an embedding close to it in the latent dimension is found, the corresponding answer returned. A direct decendant of the Question Answering task is the Visual Question Answering task[26], which provides answers to questions, each of which is based on a still image. Many solutions fuse encoded representations of a question and an associated image during training[27], where encodings often incorporate other components like LSTMs. The LSTM -Stacked Attention[28] strategy takes this solution further by stacking an attention network to allow the first attention layer to highlight an area of the image under review, so that a second attention layer can further process the highlighted area. VideoQA[4] adds an additional dimension to the challenge in that answers are predicted for questions about a video sequence. Approaches include extending the approach applied in static VQA by fusing the encoded representations of video and text while applying transformer based techniques [29], and it has been shown that static frame analysis allows good performance on VideoQA[23].

Solutions to the Question Answering family of challenges often apply supervised learning strategies, where the samples are questions and the labels are answers. Samples and labels are often generated either through human labelling or via tools that generate questions by assessing statistical distribution in text[30]. As an example, the MSRVTT-QA questions were generated by running the statistical tool over the MSR-VTT video captioning database. Accuracy metrics describe the ratio of correct responses to multiple choice or ended questions[21]. Deployments that fine-tune classifications are common in the supervised learning community, and in the QA family, specific domain solutions have been created by curating training sets[31].

Common components of QA solutions include LSTMs or transformer networks that were built over the classification capability of simple neural network connected units. Final layers generally perform a softmax process that converts the value of a logit z_x at an output x to a probability p_x, given by

$$p_x = \frac{exp(z_x/T)}{\sum_y exp(z_y/T)} \tag{1}$$

T is a temperature that can be used to adjust the contrast between softmax probabilities. Classic knowledge distillation[5] exploits that capability by extracting information on the ratios between non-maximal logits, which express better at suitably high temperatures, in order to train simpler models. Further work clarified that if the number of classes is small, probability distributions between artificially generated subclasses on a teacher model can provide useful additional information to a student model[6]. **The additional information can be expressed in bits**[7].

The video question answering task associates high dimensional samples (videos and questions) and low dimensional labels (answers). Supervised strategies give rise to the possibility that samples could be associated with more than one label, or are improperly labelled[32, 33]. Unsupervised learning therefore has an opportunity to shine, with the capability to reduce the sample dimension or identify clusters. Cluster members can be subsequently labelled. Available techniques include k-means[34] clustering, which segments a set into k groups based on distance to group centers, DBSCAN [35]clustering, which identifies groups that contain at least a specified number of members within a specified distance, and principal component analysis[36] to find the most significant components in a set. Student-t distributed stochastic neighbour embedding[37] (t-SNE) is a technique that was first applied to the visualize high dimensional data in lower, human readable dimensions by minimizing the Kullback-Leibler(KL) divergence between high dimensional data points and the low dimensional representations. It resolves the crowding problem experienced in vanilla SNE by applying a Student-t distribution so that the associated inverse square law generates larger distances. The loss minimization function manifests as forces in the lower dimension that attract similar samples, and repel dissimilar ones. Optimizations to enhance convergence include early exaggeration in the first stage, so visualizations normally appear to start with a "big bang" effect, followed by more conservative adjustments that coalesce clusters. The authors warn that appropriate parameterization is important to guarantee convergence, and that the behaviour is not guaranteed for all destination dimensions. Useful parameters to observe are the early exaggeration factor, and a perplexity that dictates the number of items that can be in a cluster. t-SNE is a useful tool in a clustering toolkit, as it's application often results in a clearer expression of clusters in the target dimension, which can inform the configuration of other techniques like k-means or DBSCAN.

3 Mood Based Learning

Supervised learning approaches remain useful for semantic correspondence but are necessarily limited by a requirement to segment data when training before knowing a full distribution of the data to be encountered by deployed models. Unsupervised learning allows good feature based segmentation of data, however in order to attach semantic meaning, subsequent labelling is normally required. Our hypothesis is that we can use unsupervised learning to distill large lakes of weakly annotated data elements into coarse grained categories that each describe the context of each data item. The extracted context should augment subsequent classification, and we test the idea in the VideoQA domain.

3.1 Design

We test the hypothesis with experimentation, however the experiment requires a method to extract relevant categories that does not depend on supervised

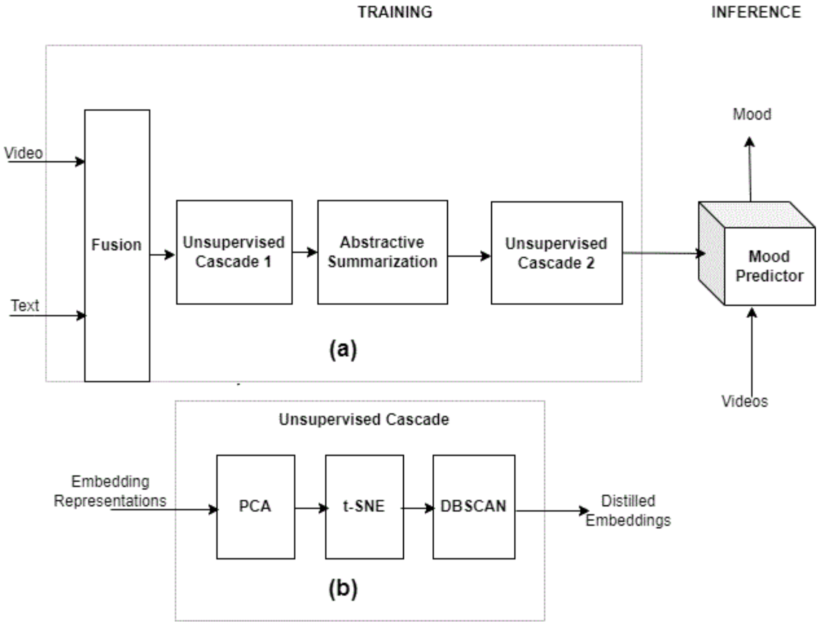

Fig. 1. Architecture, showing (a) the pipeline for automated label generation, and (b) internals of the clustering cascade used

segmentation. Our approach is to first reduce the dimension of the data to the most significant components, where the number of components is determined via the elbow technique, and then apply a cascade of clustering techniques to our dataset. To mitigate the effect of the elapsed time for clustering, we then trained a CNN model to mimic the output of the clustering set, and refer to the newly trained model as the mood model.

Additional experiments were then carried out to explore applications of the mood model. In particular, we assessed whether the context

3.2 Implementation

To validate the idea, we allow a set of coarse categorizations to emerge from the MSRVTT-QA dataset, and then subsequently train a classification model (the mood model) to accept a video stream and predict these classifications. We first embed each training video into a clustering cascade. Question text associated with each cluster member from first cascade is then summarized by cluster, and embeddings of the summaries are passed into the second cascade. Text coming out of the second cascade is futher summarized by cluster, and the those outputs are reported as human readable moods. We associate each participating video with it's mood, and the association is used to train a LSTM based model.

An overview of the infrastructure is shown in Figure 1. The structure of the LSTM-based model is shown in Figure 2.

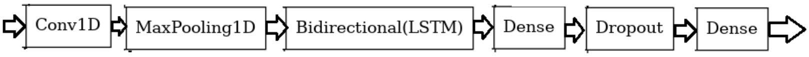

Fig. 2. Mood prediction model

A solution that intends to apply this technique for augmentation will be updated to accept an additional mood input. In order to addess the impact on the Stacked Attention LSTM network, we updated an implementation that was made available. The interface is modeled in Figure 3.

4 Experiments

Required activities included data preparation, mood extraction, mood modelling and mood deployment. We describe each of these in turn.

4.1 Data Preparation

MSRVTT-QA includes a training, validation and test set of questions and associated answers that are related to the MSR-VTT dataset. We only had access to videos in the training set however, so a decision was made to treat the training set as the full set and split it into local training, validation and test sets in the ratio 65:30:5.

For each video in the training set, a specified number of questions(default 3) are extracted for summarization. We then generate a representation for three keyframes in each video, located at the centerpoint of the video, three seconds after the beginning, and three seconds before the end. Each keyframe's representation consists of the penultimate weights predicted by an Inception model pre-trained on Imagenet, and an appended sequence that is composed of compressed background subtracted frames at 5 frames per second, three seconds prior to, and after the keyframe. We then concatenate the sequence of keyframes and background subtracted representations in reverse order. In order to reduce dimesionality we apply run-length encoding on the resulting sequence, and then prepad it with zero if shorter than 1000, or truncate the sequence at the beginning if the sequence is longer than 1000. 1000 was an arbitrary number that was in a similar ballpark as the size of the embedding created by the BERT sentence module, ie, 768. The association between each video embedding and associated text is maintained.

4.2 Mood Extraction

Two clustering cascades were applied to generate the classifications. Each cascade consists of a PCA process, a t-SNE process, and a DBSCAN process. The output from the first cascade is input into the second. For our experiment the number of components extracted by the first PCA process is set to 20, based on a grid search. The result is then applied to a t-SNE process that executes for 1000 iterations with a perplexity of 10. A grid search for the best distance to identify groups of at least 5 members is then executed, and for each of those groups, abstractive summarization is applied to the sentences associated with group members.

For each summary, an embedding is generated using the BERT sentence transformer model. The embedding is then fed into the second clustering cascade. The second cascade starts with a PCA process that filters the 3 most significant components A t-SNE process that expresses it's data in two dimensions is then applied for 3000 epochs, followed by a DBSCAN process that reports on groups of at least two.

4.3 Mood Modelling

The model to predict moods consists of an initial convolution layer followed by a max pooling layer, which then feeds into a two stacked LSTMs, the second of which is bidirectional. We then feed signals into a fully connected layer, then a dropout layer and a finally a softmax with the number of output classes set to the number of groups emitted by the final DBSCAN process. Major components of the model are shown in figure 3. For early training, the learning rateset to the default of 0.01, and exponential rate decay is applied after 50 epochs.

4.4 Mood Deployment

The mood of the environment is first extracted as a prediction from the mood module. Mood is then applied as an additional input to the SAN-LSTM network.

The experiment involved modifying a code base that implemented the LSTM Stacked-attention model, to insert an additional input for the mood, then executing training in both original and modified versions to assess differences.

5 Results.

A discussion of each assessible component in the pipeline is useful, given that the process implements a cascade.

5.1 Clustering

After the embeddings are generated, the data is compressed to 20 components using PCA. A visualization of the data represented with 1000 components as well as 20 components is shown in Figure 4 (a) and (b) respectively.

After 1000 iterations of t-SNE, the visualization is shown in Figure 4(c). The result of summarizing the group of questions that emerged from the first cluster cascade was then embedded, and passed into a second cascade. Figure 4(d) is a visualization of the result of second stage clustering.

Given approximately 12,000 data points, our experiments generated between 100 and 250 classes with a descriptive label.

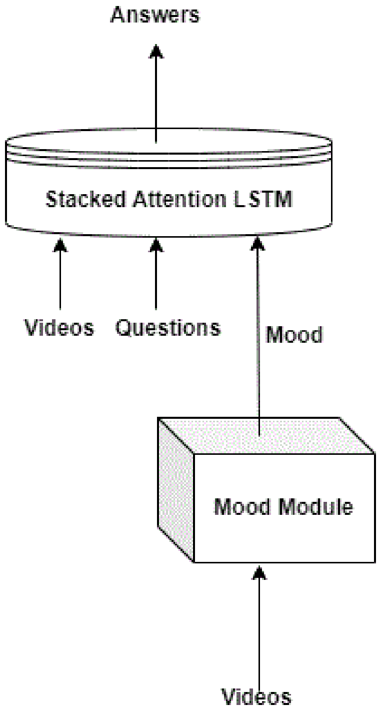

Fig. 3. Deployment Model

5.2 Model Training

It should be noted that the preferred summarization model to group the items in the cluster was the `pegasus/x-sum model`, however given the number of times the `x-sum` summarizer returned "All images are copyrighted", we were tempted forego that model and work with `pegasus/cnn-daily-mail`. Results were however comparable.

Both models consistenly had an initial flat region for approximately 10 epochs (the flat region was more pronounced for the `cnn-dailymail` generated classes) after which the training rate increased and stabilized at approximately 99% by epoch 80.

5.3 VideoQA Integration

We then modified the Stacked Attention LSTM model (SAN-LSTM) network to incorporate a mood input, and trained versions that are tuned with both the `x-sum` mood model(XSUM-MOOD-SL) as well as the `cnn-dailymail` mood model(CNN-MOOD-SL). Figure 6 shows a comparison of average training losses when comparing these models, from epoch 40 to epoch 120. Although the average

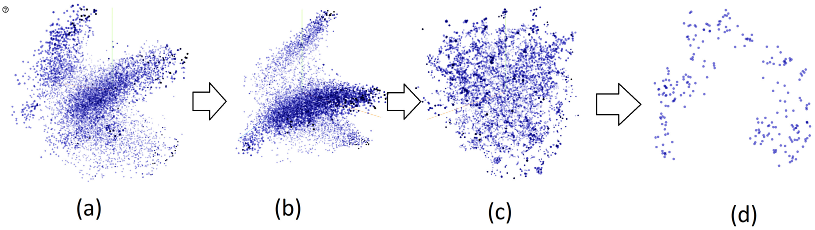

Fig. 4. Progression of the clustering technique showing (a) raw embeddings in 1000 dimensions, (b) representation after reduced to 20 dimensions by PCA (c) t-SNE for 1000 iterations, (d) remapping of updated clusters

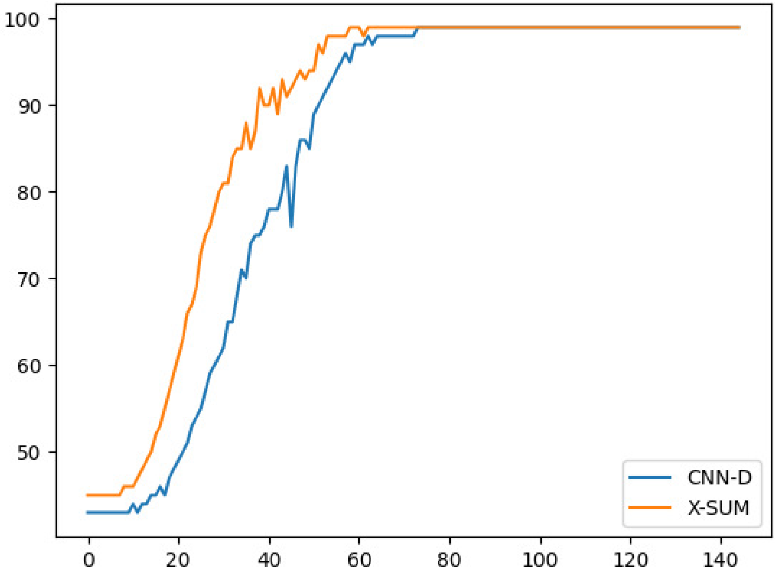

Fig. 5. Mood Model training accuracy over 140 epochs (Average of three experiments)

losses of the CNN mood model were higher for early iterations,it is possible that could have been due to initializations. Eventually, however, the losses make it clear that the augmented models outperform the vanilla models. Average results on evaluations on our adjusted sample of MSVRTT-QA are presented in Table 1.

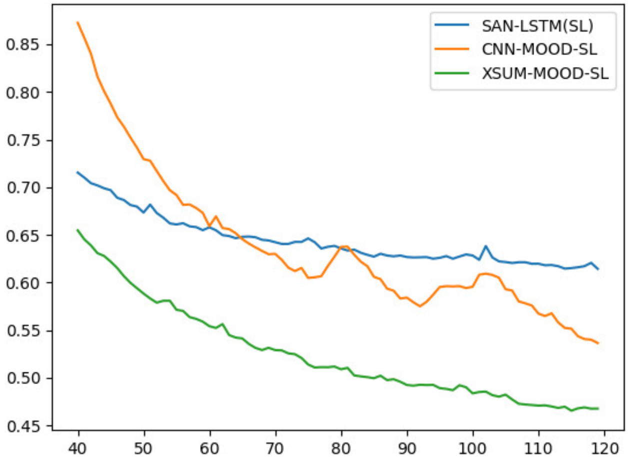

Fig. 6. Loss training Naive SAN-LSTM vs with Mood Enhancement over 120 epochs

	SAN-LSTM	CNN-DMAIL-SL	XSUM-SL
Accuracy	36%	39%	38%
Loss	0.61	0.55	0.47

Table 1:Average accuracy and loss from baseline and updated models(Average over 10 experiments)

6 Discussion

Initial concerns that labels generated from the `pegasus/x-sum` model did not match expected semantics were probably overestimated as the models appeared to perform comparably. In retrospect, given that a machine has more consideration for whether a cluster exists than the semantic significance, that was not surprising. We note however, that despite the superior loss of the `x-sum` augmented model during training, validation results show that the `cnn-dailymail` augmented model yielded marginally superior performance in accuracy, but a clear delineation in the behaviour when loss was assessed. Average loss of the vanilla model without additional inputs centered around 61%. Average loss of the augmented structure that performed better (after being trained on the pegasus/x-sum model) was 47%, This translated to a spread of 0.14, evaluating to a 23% improvement of the baseline.

Another point of discovery is that the early flat region while training the LSTM based model was unexpexted. A small source of amusement is that in the initial stages of model development we constantly interrupted training to tune hyperparameters based on the impression that the model was not learning. Our current hypotheses is that in the flat region, the model is learning how to learn to associate high dimensional data with a relativey complex output. Given that no published theory has been encountered at the time of writing that fully explains the phenomenon, we believe it could be fertile ground for future work. We suspect that inspection of intermediate layer outputs could clarify the issue.

7 Future work

The anomaly detection strategy is to be evaluated on known anomaly detection datasets to judge it's performance relative to other approaches. Also, it is recognized that design decisions made during training, such as the number of sample points in a video stream, are likely to introduce a form of quantization error. Given that the decisions were to allow sufficient data points with the available memory, the error is unavoidable. It is therefore useful to evaluate and report on the potential for error during samplling.

8 Conclusion

This paper describes an approach to create labels from data and use those labels to enhance classification. While the example application is in the video question answering domain, we believe the approach is applicable to any domain that processes data is expressed as a time sequence.

References

1. Zhou, Z., Yu, H. and Shi, H.:Optimization of Wireless Video Surveillance System for Smart Campus Based on Internet of Things. IEEE Access Special Section on Panoramic Video with Virtual Reality (2020).
2. Ali, M, et. al.,:RES: Real-Time Video Stream Analytics Using Edge Enhanced Clouds. IEEE Transactions on Cloud Computing (Volume: 10, Issue: 2, 01 April-June 2022)
3. Yadav, R. and Kumar, R. : A Survey on Video Anomaly Detection. 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 2022, pp. 1-5, doi: 10.1109/DELCON54057.2022.9753580.
4. Xu D. et. al.,:Video Question Answering via Gradually Refined Attention over Appearance and Motion. Proceedings of the 25th ACM international conference on Multimedia (2017).
5. Hinton, G. et. al.,:Distilling the Knowledge in a Neural Network. Conference on Neural Information Processing Systems Workshop (2015).
6. Muller, R. et. al.,:Subclass Distillation. Conference on Neural Information Processing Systems Workshop (2019).
7. Sajedi, A. and Plataniotis, K.,:On the Efficiency of Subclass Knowledge Distillation in Classification Tasks. Association for the Advancement of Artificial Intelligence (2022).
8. Sun, S. et. al.,:Scene-Aware Context Reasoning for Unsupervised Abnormal Event Detection in Videos, Proceedings of the 28th ACM International Conference on Multimedia (MM '20), pp. 184-192, 2020.

9. Ouyang, Y. and Sanchez, V.,:Video Anomaly Detection by Estimating Likelihood of Representations Yuqi Ouyang. 25th International Conference on Pattern Recognition (ICPR), 2021
10. X. Zhou, Y. Chen, and Q. Zhang,:Trajectory Analysis Method Based on Video Surveillance Anomaly Detection, 2021 Chinese Automation Congress (CAC), 2021
11. Deng, J. et. al.,:ImageNet: A Large-scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (2009).
12. Krizhevsky, A. et. al.,:ImageNet Classification with Deep Convolutional Neural Networks. Conference on Neural Information Processing Systems Workshop (2015).
13. Simonyan, K. and Zisserman, A.,:Very Deep Convolutional Networks for Large-Scale Image Recognition. The 3rd International Conference on Learning Representations (2015).
14. Szegedy, C. et. al.,:Rethinking the Inception Architecture for Computer Vision. IEEE Computer Vision and Pattern Recognition (2016).
15. He, K. et. al.,:Deep Residual Learning for Image Recognition. IEEE Computer Vision and Pattern Recognition (2016).
16. Rumelhart, D. and McClelland, J.,:Learning Internal Representations by Error Propagation. MIT Press- Parallel Distributed Processing: Explorations in the Microstructure of Cognition (1987).
17. Hochreiter, S. and Schmidhuber, J.,:Learning Internal Representations by Error Propagation. MIT Press- Parallel Distributed Processing: Explorations in the Microstructure of Cognition (1987).
18. Vaswani, A. et. al.,:Attention is all you need. Conference on Neural Information Processing Systems (2017).
19. Devlin, J. et. al.,:BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2019).
20. Zhang, J. et. al.,:PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. Proceedings of Machine Learning Research(2020).
21. Reddy, S. et. al.,:CoQA: A Conversational Question Answering Challenge. North American Chapter of the Association for Computational Linguistics(2019).
22. Yang, A. et. al.,: Zero-Shot Video Question Answering via Frozen Bidirectional Language Models. IEEE Computer Vision and Pattern Recognition (2022).
23. Lei J. et. al.,:Revealing Single Frame Bias for Video-and-Language Learning. IEEE Computer Vision and Pattern Recognition (2022).
24. Yang, M. et. al.,:CoQA:Knowledge-based question answering using the semantic embedding space. Expert Systems with Application(2015).
25. Saxena, A. et. al.,:CoQA:Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. Association for Computational Linguistic(2020).
26. Antol, S. et. al.,:VQA: Visual Question Answering. International Conference on Computer Vision(2015).
27. Hu, E. et. al.,:Learning Answer Embeddings for Visual Question Answering. IEEE Computer Vision and Pattern Recognition(2018).
28. Yang, Z. et. al.,:Stacked Attention Networks for Image Question Answering. IEEE Computer Vision and Pattern Recognition(2018).
29. Yang, Z. et. al.,:BERT Representations for Video Question Answering. IEEE Computer Vision and Pattern Recognition(2020).

30. Heilman, M. and Smith, N.,:Good Question! Statistical Ranking for Question Generation. Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics(2010).
31. Castro, S. et. al.,:WildQA: In-the-Wild Video Question Answering. Association for Computational Linguistics(2022).
32. Fan, J. et. al.,:WildQA: Partial Label Learning Based on Disambiguation Correction Net With Graph Representation. IEEE Transactions on Circuits and Systems for Video Technology(2021).
33. Hao, D. et. al.,:Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction and their impact on classification performance. IEEE Journal of Biomedical and Health Informatics(2021).
34. Hartigan, J. and Wong, M.:A K-Means Clustering Algorithm. Applied Statistics 28 - Blackwell Publishing for the Royal Statistical Society (1979).
35. Ester, M. et. al.,:A density-based algorithm for discovering clusters in large spatial databases with noise. Knowledge Discovery and Data Mining(1996).
36. Pearson, K.:On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of ScienceThe London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science(1901).
37. van der Maaten, L. and Hinton, G. :Visualizing Data using t-SNE. Journal of Machine Learning Research(2008).
38. Rand, W.:Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association(1971).
39. Rousseeu, P.:Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics(1971).