Dublin City University Participation in the VTT Track at TRECVid 2017

Haithem Afli[2], Feiyan Hu[1], Jinhua Du[2], Daniel Cosgrove[2],
Kevin McGuinness[1], Noel E. O'Connor[1],
Eric Arazo Sanchez[1], Jiang Zhou[1]
and Alan F. Smeaton[1]*.

[1]Insight Centre for Data Analytics,
Dublin City University, Dublin 9, Ireland

[2]ADAPT Centre for Digital Content Technology,
Dublin City University, Dublin 9, Ireland

Abstract

Dublin City University participated in the video-to-text caption generation task in TRECVid and this paper describes the three approaches we took for our 4 submitted runs. The first approach is based on extracting regularly-spaced keyframes from a video, generating a text caption for each keyframe and then combining the keyframe captions into a single caption. The second approach is based on detecting image crops from those keyframes using saliency map to include as much of the attractive part of the image as possible, generating a caption for each crop in each keyframe, and combining the captions into one. The third approach is an end-to-end system, a true deep learning submission based on MS-COCO, an externally available set of training captions. The paper presents a description and the official results of each of the approaches.

1 Introduction

TRECVid is a long-running, global benchmarking activity for content-based operations on video. Running annually since 2001, TRECVid's goals are to promote open, shared evaluation on a range of tasks [16]. A team of researchers from Dublin City University (the Insight and the ADAPT Research Centres), combined to submit 4 runs in the video-to-text (VTT) caption-generation task in the 2017 running of TRECVid. This task, described elsewhere in [1], requires participating groups to generate natural language captions for more than 1,800 videos using no external metadata, just an analysis of the video content

Our team participated in the VTT caption-generation task in TRECVid in the previous year (2016) [11], where we submitted a single run. This was based on identifying 10 keyframes per video (videos in 2016 averaged about 8s in duration, the same as in 2017) and for each keyframe we ran over 1,000 pre-trained semantic concept detectors which detected various kinds of behaviour,

*Contact author: alan.smeaton@dcu.ie

objects and locations using a VGG-16 deep CNN. We then used an open source image-to-caption CNN-RNN toolkit called NeuralTalk2 to generate a caption for each keyframe and then we combined the 10 image captions, linguistically, thus effectively using a decision-level approach.

One of our observations from last year's submission was that we generated captions which sometimes attended to the correct and sometimes the wrong object in the video/images and the resulting description of the video was a poor match to the groundtruth provided by NIST. Improving on this using image salience was one of our goals in this year's participation.

2 Submitted Runs

This year we submitted 4 runs for evaluation, outlined as follows;

- Run 1: Decision-level approach, combination system with NeuralTalk2 implemented in pytorch;
- Run 2: Decision-level approach, combination system with NeuralTalk2 implemented in tensorflow;
- Run 3: Cropping each of 10 keyframes into 10 crops based on image salience, generate descriptors for each of the 10 crops from each of the 10 keyframes and use these as input to a caption combination system with NeuralTalk2;
- Run 4: An end-to-end CNN-LSTM fine-tuned system.

We now introduce each of these in turn.

2.1 Approach 1: A Decision-Level System

Building on existing research conducted work in the areas of Computer Vision (CV) and Natural Language Processing (NLP) as well as our own submission to the 2016 TRECVid VTT task, we created a new decision level system based on the combination of different keyframe-generated captions. The architecture of this is outlined in Figure 1.

2.1.1 Background: NeuralTalk2+

As we can see in Figure 1 an open sourced image-to-caption CNN-RNN toolkit – NeuralTalk2 – was used in generating captions from the extracted images.[1] NeuralTalk2 is written in Torch and it is batched and runs on a GPU. It also supports CNN fine-tuning, which helps with improving performance. NeuralTalk2 takes an image and predicts its sentence description with a Recurrent Neural Network. Since we segmented the video into several static images, we generate one caption for each image of the video as one of the candidates for the overall video caption.

We decided that while the NeuralTalk2 architecture was still suitable for our purposes, we might benefit from changing the underlying implementation from Torch to a Python-based system. We reasoned that more popular and modern frameworks would lend themselves to easier improvement of the system.

Two alternate versions of NeuralTalk2 were submitted as TRECVid VTT runs, one written in Pytorch [2] (run1) and the other in Tensorflow [3] (run2). Both were trained on the MS-COCO dataset [10].

[1] https://github.com/karpathy/neuraltalk2

[2] NeuralTalk2 in PyTorch; https://github.com/routianluo/neuraltalk2.pytorch

[3] NeuralTalk2 in Tensorflow: https://github.com/routianluo/neuraltalk2-tensorflow

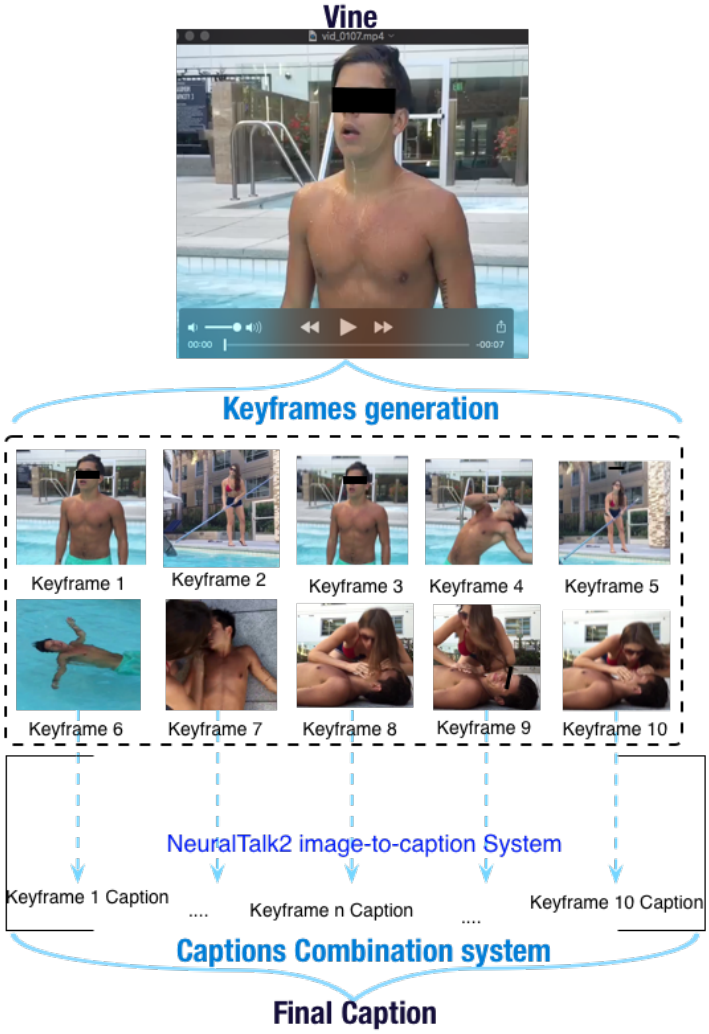

Figure 1: Architecture of our Decision-Level system

2.1.2 The Combination System

We ran the caption-generation application many times on sub-samples of the complete videos and used the system combination technique drawn from speech recognition and machine translation (MT) in which outputs from different systems are combined to generate a new final caption for submission [5]. The basic idea behind this is that a backbone will be selected based on a measure, such as the Minimum Bayes Risk (MBR). Then, all other outputs (sub-video captions) will be aligned against the backbone under a specific alignment metric, such as TER [17], Meteor [2]. Finally, a best path search will be carried out to generate the final result, i.e. the final caption for submission.

During the process of searching for the best/combined caption, many features can be used to improve the combination performance, such as the language model, posterior probability or word confidence. We use a 5-gram language model in our experiments.

In our combination scheme, each video is split into multiple keyframes. Each keyframe is then fed into the NeuralTalk2 system, and multiple image captions are generated and are regarded as candidates for combination. The potential problem in current combination systems is that in principle these individual images are different which will result in different captions. However, the MT combination system is mainly based on a statistical method for managing different translations coming from the same source sentence. Thus, in the image caption combination task, we cannot produce a result which includes different objects from different images, but a result with the objects which have high frequencies across all candidate captions. In future, we plan to use the language generation method to combine candidates from different sources, i.e. feeding all candidates into an end-to-end neural network with an attention mechanism so that it can automatically select which objects need to be included in the output.

To build the combination system, we use an open source machine translation (MT) combination toolkit: MEMT [7].[4]

2.1.3 Data for Language Modeling

The data used to train the language model includes:
- flickr30K: 158,915 sentences
- flickr8K: 40,460 sentences
- mrsVTT: 166,180 sentences
- MS-COCO Train: 415,795 sentences
- MS-COCO Val: 203,450 sentences
- UIUC Pascal Sentence: 4,997 sentences
- WMT MMT: 290,001 sentences
- flickr8k_lemma: 40,460 sentences

We use KenLM [8] to build a 5-gram language model.

2.1.4 Data for tuning and testing the combination system

We randomly select sentences from the TRECVid 2016 data set [5] to build a development set (devset) and a test set (testset). The devset includes 1,056 sentences, and the testset includes 1,057 sentences. Each video has two references.

2.2 Approach 2: Cropping keyframes

In submitted run1 and run2, we combined captions from 10 keyframes in each video to generate the final caption. In this approach, the hypothesis is that some local characters of each keyframe might not be captured while using a CNN to extract image feature representations. One way to overcome this is to generate image crops from each keyframe so that some important local patches can be captured and have a greater probability to contribute to the generation of final captions. This approach is used in run3. In order to achieve this we automatically generate 10 crops for each of the 10 keyframes for each video. Figure 2 shows an example of one keyframe and its automatically generated crops. We can see that in the crops the vehicle is appearing repeatedly, thus we can anticipate that in the combination system the corresponding concept should get some important weight making it more likely to appear in the final aggregated video caption.

These are the steps we took in order to generate the crops in run3:
- From each video we use the same 10 keyframes as in run1 and run2. Saliency maps for each keyframe are also generated using methods described in [12].

[4] https://github.com/kpu/MEMT.git

[5] The data set contains 200 videos as samples and 1,913 videos as the official test set.

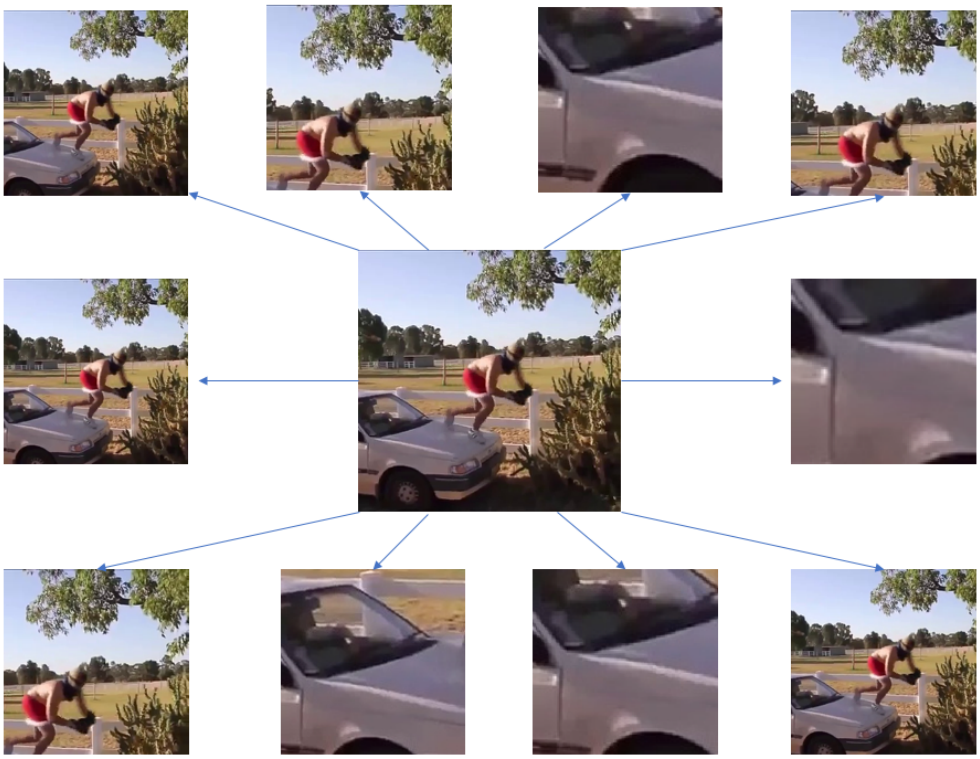

Figure 2: 10 automatically-generated crops from one example video keyframe

- Then we compute a heatmap of crops of different ratios. In the experiment we choose 10 different aspect ratios to compute heatmaps and for each of these the top 10 candidate regions in the keyframe image are chosen. After this process we in fact generate 100 crops from the original keyframe, each based around capturing as much of the salience heatmap as possible, and these crops are square.
- For the 100 candidate crops we then compute an aesthetic score similar to the method described in [9] and rank the crops based on this score. Only the top 10 crops with highest aesthetic scores are chosen as the final version of crops. The aesthetic scores are generated using a pre-trained classifier and the motivation here is to choose image crops which are visually appealing to the human eye and which capture what the human eye would believe to be the most important visual aspect(s) of the image/keyframe, even though they will never actually be viewed by a person. The classifiers are CNN pre-trained on ImageNet, fined tuned with images downloaded from the DPChallenge website [6].
- Each of these 10 top-ranked aesthetic crops are submitted to NeuralTalk2 to generate a caption. That means we have 10 valid captions for each of 10 crops for each of 10 keyframes, in a video. We use the MT Combination techniques from approach 1, to combine these together into one caption.

[6] `http://www.dpchallenge.com/`

2.3 Approach 3 - An end-to-end system

For the submitted run4 we decided to build a sequence-to-sequence model where the input is constituted by the frames from the video and the output by a sequence of words describing the frames. The model used was developed in [19] and it introduced, as a main contribution, a method to use variable length inputs and outputs for video captioning models. This is a very common practice in the field of machine translation, where the lengths of the input and the output of the model are not fixed. This allows us to train an end-to-end model for video captioning.

2.3.1 The sequence-to-sequence – video-to-text model (S2VT)

The model consists of a stack of two Long-Short Term Memory (LSTM) cells together with a pre-trained convolutional neural network (CNN) that is used to extract the features from the keyframes. This model was trained and fine-tuned on a large dataset for video captioning, and that implies that the training was time expensive.

The first part of the model consisted of a 16-layer VGG model [15] that is used to extract the features from the video frames. Figure 3 shows how this sequences of features are provided to the LSTM cells to predict the corresponding captions.

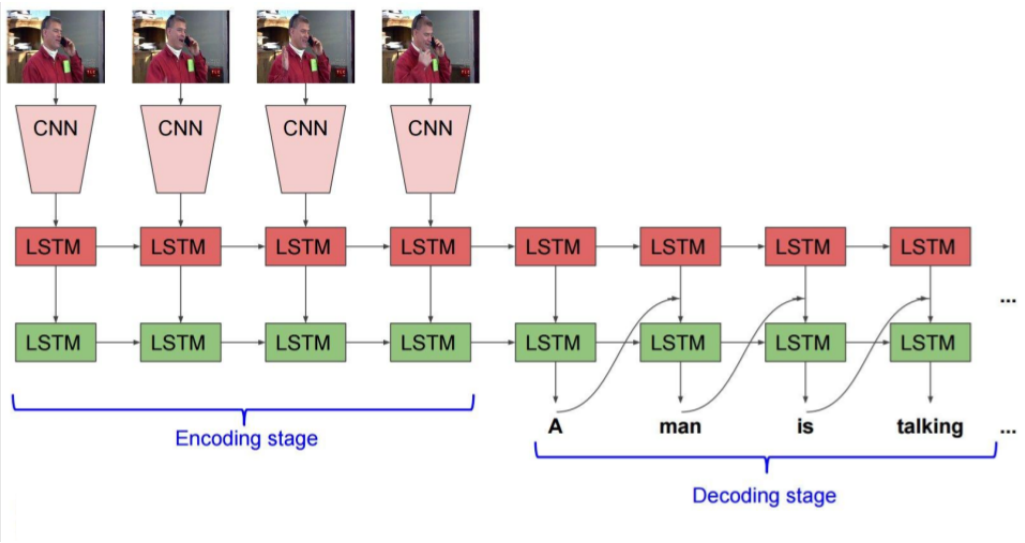

Figure 3: Sequence to sequence – video to text model from [19]

During training, the LSTM had two stages: encoding and decoding. In the encoding stage the frames are recurrently provided to the first LSTM. The hidden states are consequently updated and the output is padded with zeros and provided to the second LSTM. This zero padding will be replaced during the decoding stage by the concatenation of the output of the second LSTM (that corresponds to a predicted word of the sentence). From these frames the LSTM generates a representation of the scene that is used in the decoder stage to recursively predict the words of the caption. Each word is fed to the LSTM cell input to predict the next word.

2.3.2 Our approach

For the run4 submission we approached the task by introducing to the model, knowledge from other datasets. We performed different experiments, training models with a larger dataset to increase its performance and generalisation. The MRS-VTT 2017 dataset [20] is used for this purpose.

First, the features from the videos are extracted with the VGG-16 CNN, which is pre-trained in the ImageNet dataset [14]. Then, these features are used to fine-tune the LSTM cells initialised with the pre-trained weights provided by [19]. These pre-trained weights are fine-tuned on the MSVD dataset [3] (a standard YouTube corpus of around 2,000 videos). Other experiments have been done initialising the LSTM cells with random weights.

Figure 4 show some samples of the captions generated by the model trained on MSVD. The captions are encouraging for further development of the model.

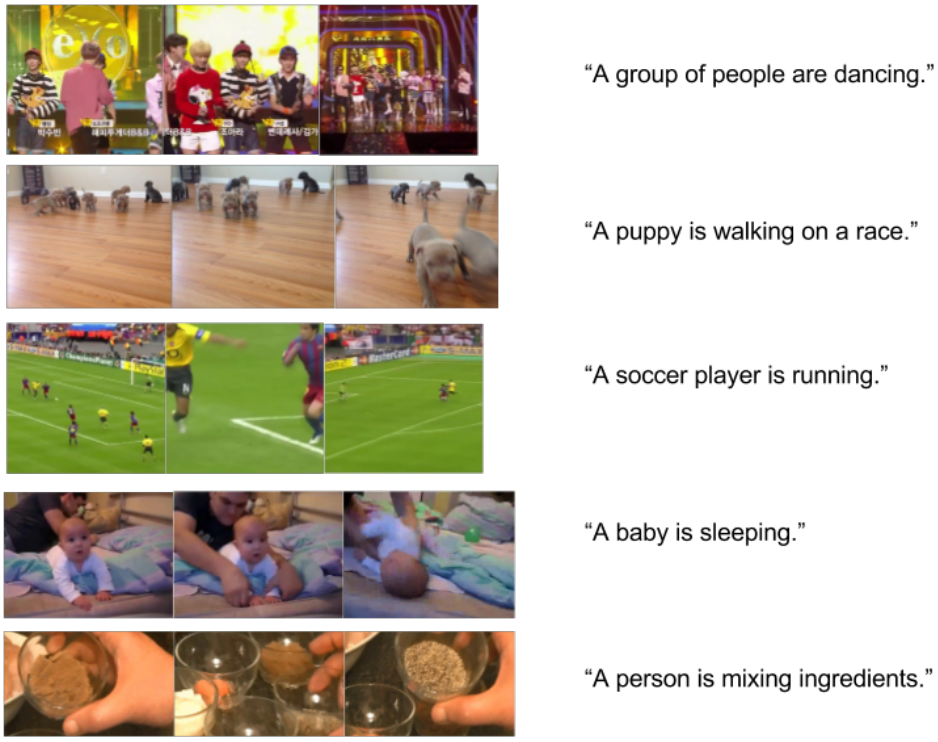

Figure 4: Captions generated with the pre-trained model provided by [19]. The three first rows correspond to videos from the TRECVID 2017 test set and the last two rows from the MRS-VTT 2017 test set.

3 Results and Performance

Tables 1 and 2 show the comparison among our 4 submitted runs. Table 1 is the result of BLEU2 to BLEU5. Our run1 demonstrates best performance in this evaluation metric while

our run2 shows constantly best performance when using METEOR 2 to METEOR 5 as an evaluation metric. We can observe that run1 and run2 achieve very similar performance overall, while run3 follows those 2 runs, which in turn outperforms run4 on almost all evaluations metrics.

Table 1: BLEU metrics of run1 to run4

	BLEU ref 2	BLEU ref 3	BLEU ref 4	BLEU ref 5
run 1	**0.0055**	**0.0080**	**0.0100**	**0.0272**
run 2	0.0055	0.0075	0.0091	0.0229
run 3	0.0044	0.0046	0.0063	0.0126
run 4	0.0030	0.0045	0.0040	0.0145

Table 2: Meteor metrics of run1 to run4

	meteor ref 2	meteor ref 3	meteor ref 4	meteor ref 5
run 1	0.1375	0.1504	0.1617	0.1928
run 2	**0.1398**	**0.1510**	**0.1630**	**0.1949**
run 3	0.1168	0.1302	0.1414	0.1705
run 4	0.1164	0.1230	0.1282	0.1463

Table 3 shows the official NIST results in term of Consensus-based Image Description Evaluation (CIDEr) [7] and CIDEr Defended (CIDEr-D) [18]. We can see that these results confirm BLEU and Meteor results where run1 and run2 achieve similar performances with a small difference of 0.001 point using CIDEr evaluation.

Table 3: CIDEr metrics of run1 to run4

	CIDEr	CIDEr-D
run 1	**0.184**	**0.122**
run 2	0.183	**0.122**
run 3	0.146	0.093
run 4	0.073	0.041

We now examine the performance of each of the runs in turn.

3.1 The Decision-Level method experiments – run1 and run2 development results

We use BLEU4 [13] to show the results for the 2 runs on the development data, described in Section 2.1, using this approach. All results are case insensitive.

3.1.1 Combination results without the language model

The combination results on our internal test set (from TRECVID2016 data) without the 5-gram language model in the combination system are shown in Table 4:

[7] https://github.com/vrama91/cider

SYS	1	2	3	4	5	6	7	8	9	comb
BLEU	4.69	4.40	4.59	4.73	4.75	4.50	**4.80**	4.58	4.24	**5.00**

Table 4: Results on the test set without a language model

where 'comb' indicates the combination system, and the numbers from 1 to 9 indicate each individual caption from the nine images generated from each input video. We can see that there is a +0.20 absolute points improvement compared to the best single system (7).

3.1.2 Combination results with the language model

Results of 9 individual systems and the combination system on the test set with a 5-gram language model are shown in Table 5:

SYS	1	2	3	4	5	6	7	8	9	comb
BLEU	4.69	4.40	4.59	4.73	4.75	4.50	**4.80**	4.58	4.24	**5.19**

Table 5: Results on the test set with a 5-gram language model

We can see that there is a 0.39 absolute points improvement compared to the best single system (7).

3.2 Results for run3 – Keyframe Cropping System

Albeit we anticipate that run 3 should at least achieve the performance of run1 and run2 if, in Tables 1 and 2 we can see that run3 doesn't perform as good as those 2 runs. The reason might be because run1 and run2 are re-trained using TRECVid 2016 dataset. Due to our limited resource we didn't have the chance to re-train a combination system using 10 crops for each keyframe. Instead we use the very same pre-trained combination system used in run1 and run2. The consequence of such an approach is that the outputs of many of our generated captions are blank.

In order to rectify this issue, generating crops for the TRECVid 2016 dataset would presumably increase the performance of this approach and this will be included in our future work.

3.3 Results for run4 – End-to-End System

To evaluate the performance of the model we used the METEOR [4] metric originally used to evaluate machine translation results. This allowed us to compare our results with the results from the original paper [19].

The data used in the experiments include:

- TRECVID 2016 : Composed of 200 videos in the training subset with two captions each. We used the training subset to validate our experiments.
- MRS-VTT 2017 [20]: a corpus of approximately 8,000 videos with 10 descriptions per video. This was used for fine-tuning and training the model.
- Microsoft Video Description corpus (MSVD) [3]: A corpus of around 2,000 videos, used in [19] to train the model.

Table 6 shows the results obtained for the different experiments performed with the model. We can see that the results reported in the original paper (in the table *S2VT*) outperform all of the other results achieved in our experiments. The same model shows a dramatic decrease in performance when is evaluated on the TRECVid 2016 dataset (*Pre-trained* in the table). This might be due to the nature of the TRECVid 2016 dataset which is considerably small compared

with the other datasets and only presents two captions per video. The decrease in performance of the model suggests that the model does not adapt well to new datasets.

Model	METEOR (%)
S2VT	29.2
Pre-trained	12.7
Fine-tuned	**13.3**
From scratch	12.4

Table 6: Results of the sequence-to-sequence – video-to-text model. S2VT corresponds to the performance reported in [19]. *Pre-trained* corresponds to the model provided by [19] and evaluated on the TRECVid 2016 test subset. *Fine-tuned* corresponds to the model trained on MRS-VTT 2017 and initialised with the weights provided by [19]. And *From scratch* corresponds to the model trained on MRS-VTT 2017 initialised with random weights.

At the same time we can appreciate that the model pre-trained in the MSVD dataset and fine-tuned in MRS-VTT 2017 is the one providing the best results, outperforming the results of the model provided in [19]. We see as well that the model trained from scratch (using random initial weights) achieves a similar performance to the pre-trained model.

3.4 Comparison to Other Submissions

The results of our submissions in comparisons to submissions from other sites puts us at about mid-table of the 13 sites who submitted for this task and is shown in Figure 5. Elsewhere the relative merits of each of the evaluation metrics is discussed [1]

CIDEr	METEOR	BLEU	STS	DA
RUC_CMU	RUC_CMU	RUC_CMU	RUC_CMU	RUC_CMU
mediamil	mediamil	mediamil	INF	NII_Hitachi_UIT
INF	INF	TJU	mediamil	mediamil
TJU	DCU	UTS_CAI	NII_Hitachi_UIT	INF
UTS_CAI	TJU	INF	TJU	VIREO
VIREO	VIREO	DCU	UTS_CAI	UTS_CAI
NII_Hitachi_UIT	UTS_CAI	VIREO	VIREO	TJU
ARETE	KU_ISPL	NII_Hitachi_UIT	CCNY	DCU
DCU	SDNU_MMSys	SDNU_MMSys	SDNU_MMSys	CCNY
SDNU_MMSys	NII_Hitachi_UIT	CCNY	KU_ISPL	ARETE
CCNY	ARETE	ARETE	DCU	KU_ISPL
KU_ISPL	CCNY	KU_ISPL	ARETE	SDNU_MMSys
UPCer	UPCer	UPCer	UPCer	UPCer

Figure 5: Ranking comparison of performance of preferred runs from each site using all metrics.

Of particular interest is the metric known as Direct Assessment, introduced in [6]. This measure is presented as a computation of average DA score from crowdsourced assessments of each caption, in the range $[0 \ldots 100]$ for each system, which are then micro-averaged per caption then the overall

average for the run is calculated. Comparing our DA score for our preferred run (run4), is shown in Figure 6 and shows that the human captioning, submitted as a benchmark to measure against, is clearly better than any automated system and while once again we rate at about mid-table in ranking among systems, the absolute differences in DA scores across the sites, is not large.

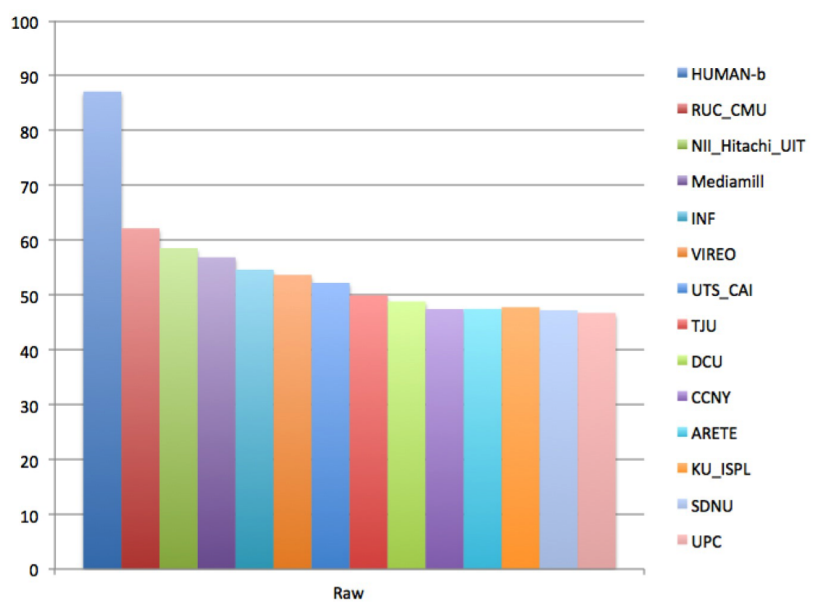

Figure 6: Direct Assessment performance results for preferred runs from each site.

4 Conclusions and Future Work

This paper reports the work done for 4 submissions for the video-to-text task in TRECVid 2017. We adopted 3 approaches based on turning videos to keyframes and generating captions for each keyframe and then combining these captions into one overall caption for the video (run1 and run2), applying salience detection to keyframes to drive an automatic cropping system which yields 10 aesthetic crops per keyframe. These crops are used to generate captions which are combined, and finally and end-to-end solution based on video in / caption out.

The performance of our submissions in comparison to others' is about mid-table across 5 different metrics for our preferred run. We are pleased with the outcome of our submissions which do not constitute entire approaches to caption generation but rather elements or "smarts". These smarts are shown to work, now what we have to do is see how well they combine with others.

Acknowledgement: The work reported here is based on research conducted with the support of Science Foundation Ireland under grant numbers SFI/12/RC/2289 (Insight Centre) and SFI/13/RC/2106 (ADAPT Centre).

References

[1] George Awad, Asad Butt, Jonathan Fiscus, David Joy, Andrew Delgado, Martial Michel, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Maria Eskevich, Roeland Ordelman, Gareth J. F. Jones, and Benoit Huet. TRECVID 2017: Evaluating Ad-hoc and Instance Video Search, Events Detection, Video Captioning and Hyperlinking. In *Proceedings of TRECVID 2017*. NIST, USA, 2017.

[2] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, 2005.

[3] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011.

[4] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

[5] Jinhua Du, Yifan He, Sergio Penkale, and Andy Way. MaTrEx: The DCU MT System for WMT 2009. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 95–99, 2009.

[6] Yvette Graham, George Awad, and Alan F. Smeaton. Evaluation of automatic video captioning using direct assessment. *CoRR*, abs/1710.10586, 2017.

[7] Kenneth Heafield and Alon Lavie. Combining machine translation output with open source: The carnegie mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, (93):27–36, 2010.

[8] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August 2013.

[9] Feiyan Hu and Alan F. Smeaton. Image aesthetics and content in selecting memorable keyframes from lifelogs. In *Proceedings of Multimedia Modelling (MMM), LNCS 10704*, pages 1–12. Springer International, 2018.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] Mark Marsden, Eva Mohedano, Kevin McGuinness, Andrea Calafell, Xavier Giro-i Nieto, Noel E O'Connor, Jiang Zhou, Lucas Azavedo, Tobias Daudert, Brian Davis, Manuela Hurlimann, Haithem Afli, Jinhua Du, Debasis Ganguly, Wei B. Li, Andy Way, and Alan F. Smeaton. Dublin City University and Partners' Participation in the INS and VTT Tracks at TRECVid 2016. In *Proceedings of TREVid, NIST, Gaithersburg, Md., USA*, 2016.

[12] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto. SalGAN: Visual Saliency Prediction with Generative Adversarial Networks. *arXiv preprint arXiv:1701.01081*, 2017.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[16] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation Campaigns and TRECVid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[17] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231, 2006.

[18] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575. IEEE Computer Society, 2015.

[19] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to Sequence – Video to Text. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[20] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, June 2016.