A Two-stage Transfer Learning Approach for Storytelling Linking

Hao Wu[1], Gareth J. F. Jones[2], François Pitié[3], and Séamus Lawless[4]

[1,4]ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin, Ireland
[3]ADAPT Centre, School of Engineering, Trinity College Dublin, Ireland
[2]ADAPT Centre, School of Computing, Dublin City University, Ireland
hao.wu@adaptcentre.ie, seamus.lawless@scss.tcd.ie, gareth.jones@dcu.ie, pitief@tcd.ie

October 2018

Abstract

This paper provides an overview of our participation in the TRECVID 2018 Storytelling Linking task. Our approach uses a RNN based neural network to learn a semantic representation of text (news topics), images and videos (collected from Twitter and Flickr posts) in the same latent space. We applied a two-stage (pre-train + fine-tuning) learning architecture to train and adjust the model (using Flickr30k and labels from online search as additional data). During the search phase of the task we take a different strategy to generate five different runs by leveraging video-length normalization and controlling the training source.

1 Introduction

The Social-Media Video Storytelling Linking task (LNK) at TRECVID 2018 [1] required participants to illustrate a news story using multimedia social media content. Starting from a news story topic and a stream of social media video and images, the goal was to link a story segment to image and video material, while also preserving the narrative flow of the overall visual story.

By being able to visually present a news topic with social media content, this task seeks to develop the technology to provide a valuable solution to people working in industries such as social media, news press, live events or even broadcasting.[1]

The task organizers released 96,590 images with no video as the development dataset. This was accompanied by 149 text queries (story segments) and an associated human labeled relevance judgment file (QRel). The test dataset consists of 93,832 images and 14,275 videos with 118 text queries. All the data provided focuses on two events: The Edinburgh Festival and Le Tour de France. The collected data include news stories from verified news sources which are used as the story topic and segments. The social media data was collected from Twitter and Flickr, and was obtained using a focused crawler to collect event specific images and videos [6].

In previous years the aim of the linking task at TRECVID was to retrieve video segments given a text query, which is quite different from the 2018 storytelling linking task. There were 2 major challenges with this new task. First, the social media content was not labeled, which makes it difficult to correctly link it to news topics. Second, we need to leverage between the relevancy and consistency of the visual story since both of them will be taken into account during evaluation. Here relevancy means whether the retrieved visual illustration is semantically related to the query and consistency cares about the relation between each visual illustration within the same topic.

The two main contributions of our proposed approach are as follows:

- We developed a RNN based neural network which can capture and combine both word-level and sentence-level information to generate text embeddings. This approach outperforms other architectures in our source domain.
- We developed a two-stage transfer learning approach which can collect more specific information about the two events (Edinburgh Festival and Le Tour de France) from the source domain. This builds upon the use of image captioning datasets, such like Flickr30k and MSCOCO, which are widely used in TRECVID tasks.

The remainder of this paper is organized as follows: section 2 elaborates upon the approach we took to addressing the task, from data processing to system design; section 3 describes the configurations of the five submitted runs and discusses the main differences between them while section 4 shows the performance of our approach on the test set.

2 System detail

In this section we describe the technical details of our system. Our approach is based on a text to image retrieval neural network model, where we use text

[1] https://www-nlpir.nist.gov/projects/tvpubs/tv18.slides/tv18.lnk.slides.pdf

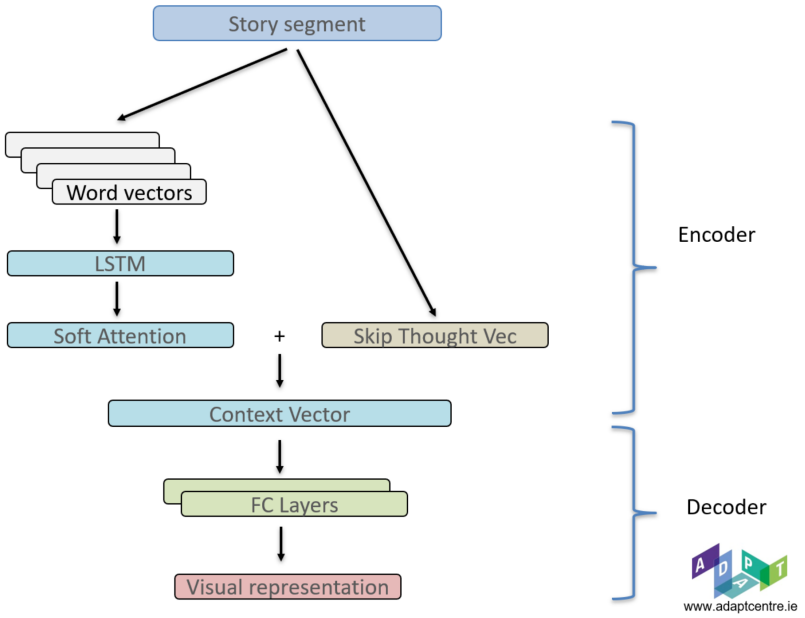

Figure 1: Model overview

(story segment) as input to predict visual features.

2.1 Data processing

Since the given test data set consists of both images and videos, here we can easily extract visual information from an image but usually a video has many scenes which may contain different information. Our solution here is to first segment a video and make each segment as semantically independent as possible, then extract visual features from the segments as our video representation. For video segmentation, a video is sliced into several clips using Shot Boundary Detection. Which is a well developed technique that automatically detects the transitions between shots in digital video, with the purpose of temporal segmentation. Particularly we use the implementation provided in FFmpeg[2]. Each generated segment is then represented by averaging the frames contained in it. Frames are uniformly sampled from the video with an interval of 0.5 seconds.

2.2 Visual embedding and text embedding

The use of pre-trained convolutional neural networks for visual content analysis [2] has become common practice in research, and we adopt this approach. We use the ResNet152 model [3] pre-trained on the full ImageNet dataset with over

[2]https://github.com/FFmpeg/FFmpeg

10 million images and 10 thousand classes. Specifically, for each object we take the output from the last pooling layer of ResNet152 as its feature vector.

With regard to text embedding, in this task we directly use the full sentences of each topic segment as our query rather than keywords. In order to capture both temporal and global information about the sentence, we combine word level embedding with sentence-level embedding to generate our text representations. The approach we use for sentence level embedding is called Skip Thought Vectors, which has demonstrated good performance in many NLP tasks including Image-Sentence Ranking [5]). Due to the small vocabulary size of the topic corpus provided in this task, it is not realistic to train text embedding from scratch although it may help us gather event specific information. Naturally, we turn to pre-trained models to generate our text embedding. For word-level embedding we use the Word2Vec model pre-trained on the Google news corpus, for sentence embedding we use the Skip Thought Vectors pre-trained model provided on Github[3] which is trained on the MovieBook and BookCorpus[4] datasets.

Finally we define $\mathbf{I}$ as our visual collection which contains all visual vectors of both image and video segments. During search, given a query, we first map it onto the visual space, then compute the cosine similarity score for every object in $\mathbf{I}$.

$$S_c = \frac{v \cdot v_q}{|v| \cdot |v_q|}, v \in \mathbf{I} \tag{1}$$

Where S_c means cosine similarity score, v_q is the visual space projection of the query. Also for video, we assume the score of a video to be the highest score of its segments. Note that this process will be elaborated in section 2.6.

2.3 Model structure

Figure. 1 illustrates the encoder-decoder structure of our proposed neural network. Inspired by the idea in Word2VisualVec, which applied a two-layer neural network for image-text retrieval task [2]. The basic idea of our model is: summarize the contextual information of a given query then map it onto the ResNet152 visual space, in which we can directly compute the similarity score with all objects in $\mathbf{I}$.

2.3.1 Encoder

The encoder combines the output of a Long Short-Term Memory (LSTM) with the sentence embedding to summarize the contextual information of the query. Given an input sentence $\mathbf{S}$ with length n, we first embed each word into a 300 dimensional text vector to get our sequence input $\mathbf{x} = (x_1, \ldots, x_n)$. We then feed them sequentially into a LSTM module with soft attention mechanism on

[3] `https://github.com/ryankiros/skip-thoughts`
[4] `http://yknzhu.wixsite.com/mbweb`

the top to generate our context vector **c**. **c** is the same size as our LSTM units, which is 1024 in this experimental setup. In the meantime, we also embed the whole sentence **S** using a pre-trained Skip Thought Vector model to get a 4800 dimensional sentence level representation **st**. Finally, we simply concatenate **c** with **st** to get the output of the whole encoder **u**:

$$\mathbf{u} = \mathbf{c} \| \mathbf{st} \tag{2}$$

where **u** captures information from both a word-level and sentence-level perspective.

2.3.2 Decoder

u is decoded through two fully connected layers into the visual space, each fully connected layer is followed by the Batch Normalization [4] and ReLU activation which is the common approach to prevent overfitting and speed up training.

$$\mathbf{fc1} = \text{ReLU}(BN(W_1\mathbf{u} + b_1)) \tag{3}$$

$$\mathbf{v} = \text{ReLU}(BN(W_2\mathbf{fc1} + b_2)) \tag{4}$$

Where **v** stands for visual vector which is the model's raw output, *BN* indicates the batch normalization.

2.3.3 Training configuration

During training, we use mean square error (MSE) as our objective function, with the following setting learning rate=0.001, decay weight=0.9 and ϵ=10^{-6} for the optimizer **RMSprop**[5].

We also apply dropout in both the encoder and decoder with a fixed dropout rate=0.2 to prevent over-fitting.

2.4 Pre-training

For both the development data and test data described in the introduction, the social media content (images & videos) which is provided has no associated text descriptions or annotations. In order to address this lack of appropriate training data, we first pre-train our model with high quality text-image pairs from **Flickr30k** [7, 8]. **Flickr30k** is a popular benchmark dataset, where each image is associated with five crowd-sourced English text descriptions.

Having completed this pre-training, we expect our model to be able to master general real-world concepts and satisfy simple queries which require no event-specific knowledge (e.g. *Pizza, Playful dogs, People having a meal*). However, the LNK task in which we are participating is focused on two specific events: *The Edinburgh Festival* and *le Tour de France*. For queries that contain event-specific terms (e.g. *Deep time Show, Museum of Edinburgh, Highlights of Chris Froome*) our model requires more specific information.

[5] `https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`

2.5 Fine-tuning

We adopt two main strategies to collect additional information specifically for the LNK task.

First, we match the **Flickr30k** corpus with keywords from the topic segments of the events, provided as part of the task. For the two events, we identified 8747 (*The Edinburgh Festival*) and 5233 (*le Tour de France*) items in **Flickr30k** which contain specific information about the events.

Second, we collect labels from the image search engines Google & Bing, using the **topic segments** + **event name** as a query (e.g. deep time show edfest). Specifically, we collect the first 10 images returned by each search engine as our labels.

Finally, we over-sample the collected target information (image-sentence pairs) using our model to expect improvement on those queries which require event-specific information.

2.6 Search

During the retrieval stage, for the ith story segment, we first pass the query to our trained model to get the model raw output. In order to preserve the consistency of the visual story we linearly combine this raw output with the modified output from the last time step *i-1* to compute R_t, which is the modified output at the current time step t. Then we directly compute the cosine similarity between R_t with all visual vectors in $\mathbf{I}$ and use the score to rank $\mathbf{I}$ to generate our result.

$$R_t = 0.2R_{t-1} + 0.8M_t \tag{5}$$

where M_t is the model raw output at time step t and R_0 is 0.

3 Submission

We submitted five runs. The main difference between them is the value of λ, which measures the video length penalty and will be used when computing the cosine similarity score.

$$\bar{S_c} = S_c \sigma \left(\frac{\lambda}{\log_2 L} \right) \tag{6}$$

Where σ is the sigmoid function; L is video length, specifically, the number of segments in the video; $\bar{S_c}$ is the modified cosine similarity score and S_c is the original score. Additionally the model of Run1 used the labels collected from both Bing and Google image search where other runs only used labels from Google.

Conf	Run1	Run2	Run3	Run4	Run5
λ	3	5	12	20	50

Table 1: λ values for the five runs.

The λ configurations of the five runs are listed in Table 1.

4 Results

According to the task instructions[6], the overall summary quality is given by the expression which is also the main metric for the evaluation:

$$\bar{Quality} = \frac{1}{N}\sum_{i=1}^{n} pairwiseQuality(i, i-1) \tag{7}$$

$$pairwiseQuality(i, i-1) = 0.4 \cdot s_{i-1} + 0.2 \cdot s_i + 0.2 \cdot t_i + 0.2 \cdot s_{i-1} \cdot s_i \tag{8}$$

Where s_i stands for relevance score of segment i and t_i is transition score between segment i and segment i-1 which correspond to the consistency.

Figure. 2 shows the performance of each team on the test data set. Run files named as "ed17" are ours while "ns" means NOVA Search team, here the NOVA Search team also submitted a manual run file for each event which is labeled in red. Overall our runs lead the evaluation in the task.

5 Conclusions and Further Work

In this paper we introduced an approach that explored use of textual and visual features, combined with a two stage transfer-learning strategy for solving the storytelling linking task in TREC 2018. More specifically, the network performed well in terms of mapping textual features into visual space and our training approach proved to be effective in the task. This provides a general solution for other event based retrieval task.

When reviewing our participation, there remain two potential improvements that could be made to our method as part of future work: First, currently we only consider the visual feature similarity for the story consistency, inspired by NOVA Search team we could make use of relevance file from the development data set and build a classifier to address this problem. Second, the image-text pairs collected from online search engines are noisy, by developing a filter we could potentially improve the quality of the data thus achieve a higher relevance score.

[6] https://www-nlpir.nist.gov/projects/tv2018/Tasks/lnk/

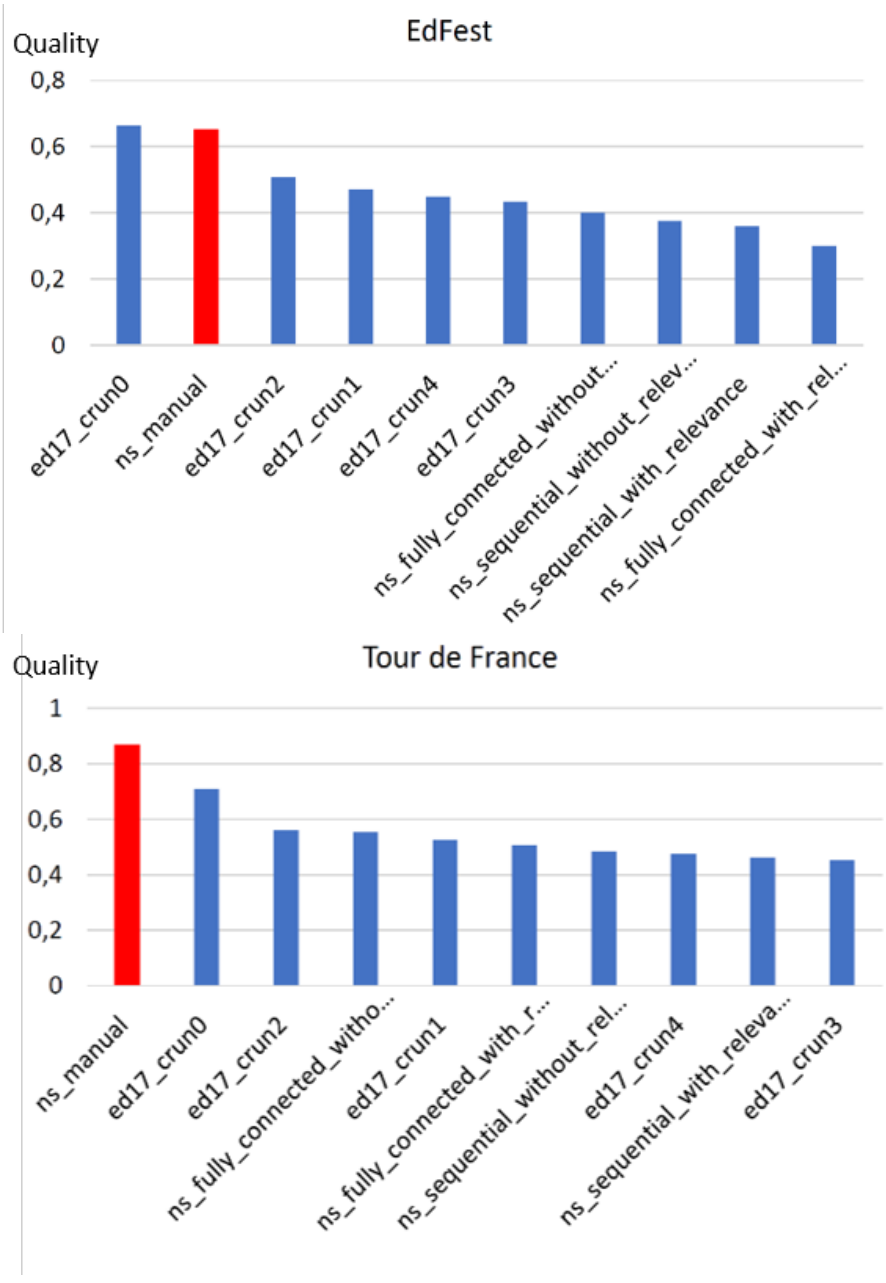

Figure 2: Result on test set

References

[1] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, David Joy, Andrew Delgado, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Joao Magalhaes, David Semedo, and Saverio Blasi. Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In *Proceedings of TRECVID 2018*. NIST, USA, 2018.

[2] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Word2visualvec: Cross-media retrieval by visual feature prediction. 2016.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[5] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[6] Joao Magalhaes, David Semedo, and Saverio Blasi. Trecvid 2018 linking task: Social media video storytelling. 2018.

[7] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[8] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. volume 2, pages 67–78, 2014.